3/6

Grondbeginselen van de digitale techniek

Inhoud

3/6.1 **Inleiding**
(verschenen in de 10e aanvulling)

3/6.2 **De getalstelsels**
(verschenen in de 10e aanvulling)

3/6.3 **Schakelingen van digitale functies** 1)

3/6.4 **De booleaanse algebra** 1)

3/6.5 **De KV-diagrammen**
(verschenen in de 14e aanvulling)

3/6.6 **De nieuwe symbolen voor digitale logika**
(verschenen in de 14e aanvulling)

3/6.7 **Bistabiele elementen (flip-flop's)**
(verschenen in de 15e aanvulling)

3/6.8 **Digitale codes en omzetters**
(verschenen in de 15e aanvulling)

3/6.9 **Digitale telschakelingen**
(verschenen in de 16e aanvulling)

3/6.10 Functie en toepassing van schuifregisters
(verschenen in de 17e aanvulling)

3/6.11 Digitale rekenschakelingen
(verschenen in de 18e aanvulling)

3/6.12 Samenstelling en werking logika-families [1)]

3/6.13 Opbouw, werking en toepassing van digitale geheugens
(verschenen in de 21e en 48e aanvulling)

3/6.14 Binaire multipliers
(verschenen in de 36e aanvulling)

3/6.15 Werking en principes van flash-geheugens
(verschenen in de 44e aanvulling)

3/6.16 Digitale perifere drivers
(verschenen in de 46e aanvulling)

3/6.17 Toepassen van geheugen-modulen in computers
(verschenen in de 49e aanvulling)

3/6.18 Digitale comparatoren
(verschenen in de 50e aanvulling)

3/6.19 Werking en principes van PLD's
(verschenen in de 51e aanvulling)

3/6.20 Werking en principes van transceivers
(verschenen in het 2e basiswerk)

3/6.21 Het transport van digitale signalen
(verschenen in de 74e aanvulling)

3/6.22 Werking en principes van Zero-Power SNAPHAT geheugens
(verschenen in het 2e basiswerk)

3/6.23 Werking en principes van clock-drivers/generatoren
(verschenen in de 83e aanvulling)

3/6.24 Werking en principes van bus-schakelaars
(verschenen in de 85e aanvulling)

3/6.25 DAS, Data Acquisitie Systemen
(verschenen in de 104e aanvuling)

3/6.26 Werking en principes van monostabiele multivibratoren
(verschenen in de 105e aanvulling)

3/6.27 Werking en principes van FIFO's
(verschenen in de 106e aanvulling)

[1] Dit hoofdstuk heeft een eigen inhoudsopgave

3/6.1

Inleiding

Digitalisering is in bijna alle belangrijke deelgebieden van de electronica doorgedrongen en wint in steeds sneller tempo terrein. Het ligt dan ook voor de hand, dat in het kader van dit hoofdstuk van 'Hobby-Elektronica' een theoretische, zij het sterk op de praktijk betrokken beschrijving niet mag ontbreken. De principes en theorieen waar bij de digitaaltechniek van uit wordt gegaan komen aan de orde, voorzover zij voor de hobbist van belang zijn. Ook worden componenten, schakelingen en functies aan een beschouwing onderworpen.
Digitale schakelingen vormen een ideaal leergebied, omdat de meeste besproken principes met eenvoudige middelen zijn na te bootsen, hetgeen ik dan ook van harte aanbeveel. In de handel zijn voor redelijke prijzen 'breadboards' te koop, voorzien van rijen contacten, met de voor digitale schakelingen gebruikelijke rasterafstand. Op deze 'breadboards' kan men componenten inprikken en zonder solderen met draadeindjes met elkaar verbinden. Sinds ikzelf een dergelijk 'breadboard' gebruik, spaar ik alle afgeknipte componenten draadjes van de door mij gebouwde schakelingen! Echter ook een stukje gaatjesprint met een aantal IC-voetjes is goed te gebruiken. De bij de experimenten benodigde IC's zijn alle uit de standaard 74xx en 40xx serie. Zij zijn over het algemeen niet duur en altijd handig om in de rommelbak te hebben, als u ze niet onmiddellijk voor de bouw van een schakeling nodig hebt. Onontbeerlijk is natuurlijk een goede lab-voeding, die tussen 5V en 12V kan worden ingesteld en een stroom kan leveren van ca. 1A. Verder is een voltmeter nodig. Een oscilloscoop is een zeer praktisch hulpmiddel om de timing van signalen te kunnen bekijken. De afgedrukte tijddiagrammen zijn naast een verduidelijking een aansporing om een en ander zelf na te meten.

Laten we nu de minder draagkrachtigen, die niet over een oscilloscoop beschikken vallen? Geenszins. In de meeste gevallen is de frequentie van de signalen zover terug te brengen, dat het met behulp van LED's mogelijk is het niveau en de verandering daarin te volgen. Een blokgolf generator is echter wel noodzakelijk. Een eenvoudige schakeling om er zelf een te maken krijgt u.

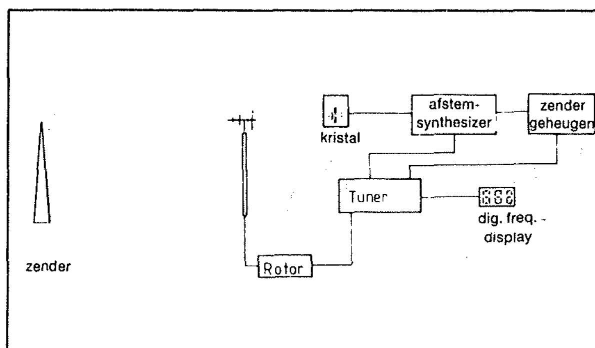
Het begrip 'digitaal' komt uit het latijns en betekent letterlijk 'vingers'. De betekenis die we er in de digitale signaalverwerking aan geven duidt ook op het feit, dat onze voorouders op hun vingers rekenden. Bij vertaling in zinsverband wordt in het latijn met digitaal

6.1 Inleiding

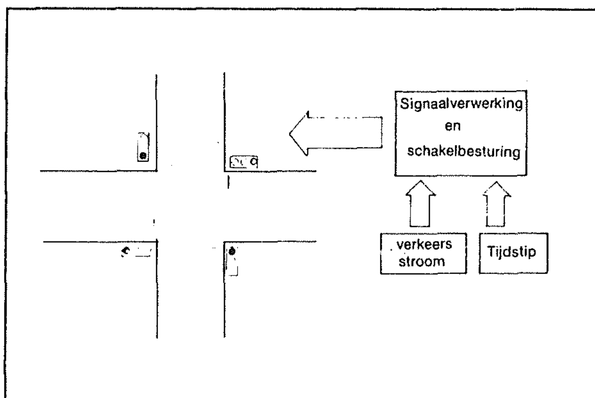
meestal cijfers bedoeld. In de digitaaltechniek en in computers worden signaalniveaus of combinaties daarvan uitgedrukt in cijfers. Bijvoorbeeld een tijd in uren en minuten of een afgelegde afstand in kilometers. Om een omzetting te verkrijgen zou het denkbaar zijn om tien verschillende spanningsniveau's te gebruiken.
Dit is in de electronica echter tamelijk complex. Hoewel dit in de toekomst misschien ooit nog eens gebeurt, wordt in de huidige techniek slechts met twee spanningsniveaus gewerkt. De ene spanning wordt aangeduid met H, hoog, high, +, ed. de andere met L, laag, low, aarde ed. In dit hoofdstuk zullen we consequent H en L gebruiken om een hoog resp. laag niveau aan te duiden. De aanduidingen '0' en '1' corresponderen in de digitale techniek niet noodzakelijkerwijs met L resp. H.

Als een digitale schakeling niet naar behoren functioneert, moet men altijd met een voltmeter of oscilloscoop het spanningsniveau van de in- en uitgangen controleren.
Natuurlijk beginnen we met zeker te maken dat de voeding en de aarde goed zijn. Het principe is tamelijk eenvoudig als een uitgang niet het juiste niveau heeft, terwijl alle condities, die het niveau van die uitgang bepalen wel goed waren, dan verwijdert men de belasting van de verdachte uitgang. Is het niveau nu wel goed, dan ligt het probleem in de belasting (bijv. kortgesloten ingang van een volgende poort), anders is de uitgang zelf defect.

Omdat er slechts twee toestanden voorkomen (H en L) noemt men een digitaal systeem binair.



Figuur 3/6.1-1: Gebruik van digitale componenten in een tuner.

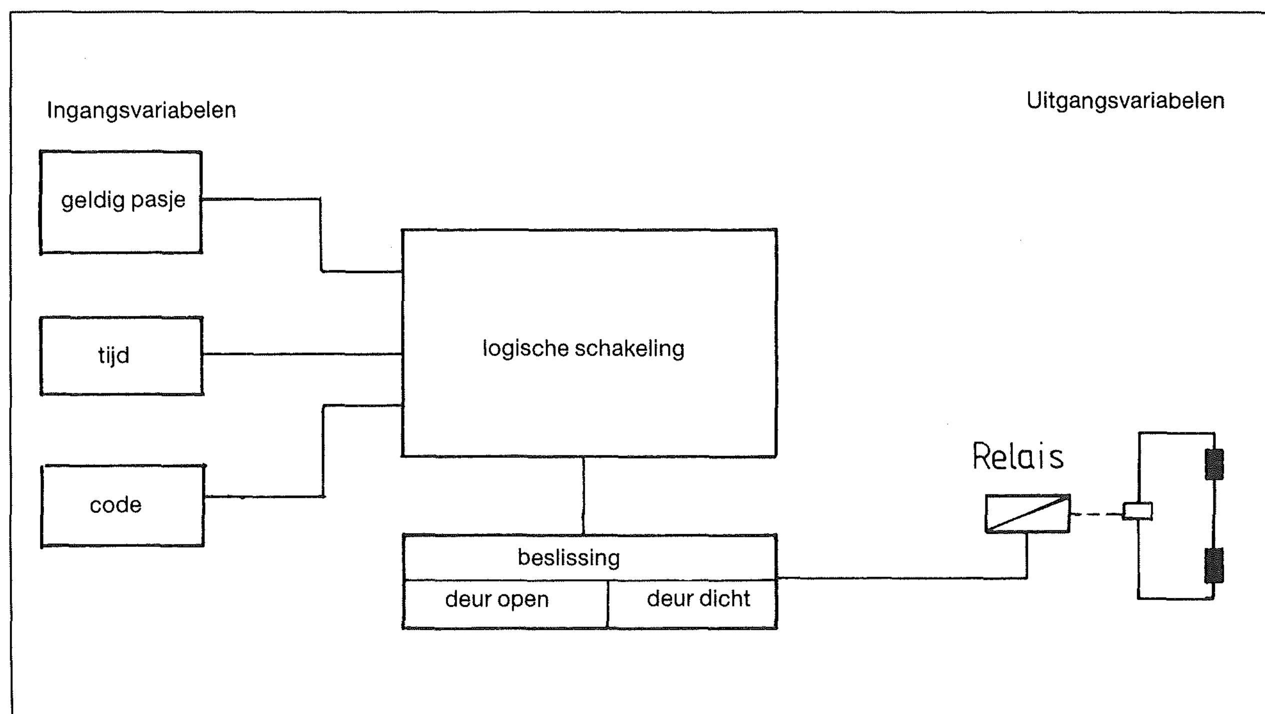


Figuur 3/6.1-2: Digitale besturing van verkeerslichten.

Het gehele thema 'digitale principes' wordt in een logische volgorde uitgelegd. Evenals bij de wiskunde is het grondprincipe van de digitale techniek het 'een plus een is twee', d.w.z. de getallensystemen en regels zoals bijvoorbeeld het tweetallig stelsel en hoe daarin op te tellen, af te trekken, te vermenigvuldigen en te delen.

De volgende stap gaat over de verschillende grondelementen of de digitale bouwstenen. De heden in gebruik zijnde bouwstenen zullen uitvoerig worden besproken. De logische functies worden uitgelegd. Schrik niet van het grote aantal. Veel bouwstenen zijn complementair, dat wil zeggen dat zij met

6.1 Inleiding



Figuur 3/6.1-3: Toegangscontrole met behulp van digitale schakelingen

andere grondelementen samen te stellen zijn. Schema's zullen in dit deel de nodige verduidelijking geven.

In het volgende deel worden uit bouwstenen schakelingen gemaakt. Door verbinden van verschillende bouwstenen wordt een schakeling samengesteld, die een vooraf gedefinieerde functie vervult. Dit wil zeggen, dat bouwstenen zo met elkaar worden verbonden, dat een van te voren bekende logische combinatie van ingangssignalen een gewenste uitgangstoestand oplevert. Een hulpmiddel bij het ontwerpen en plannen is de 'Boleaanse algebra'. Daarmee is het mogelijk zonder veel experimenteren de gewenste uitgangstoestand te berekenen.

De gebruikte rekenregels brengen het geheel terug tot de kleinst mogelijke variant van de logische functie, waaruit de logische schakeling met het minste aantal bouwstenen en verbindingen op eenvoudige wijze is af te leiden.

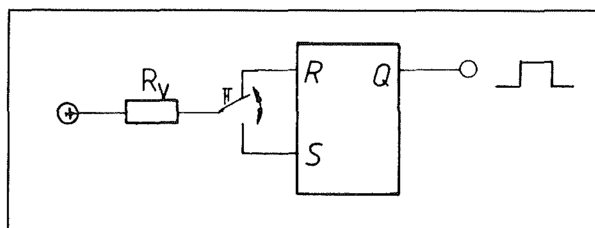
Zoals ieder uit de praktijk weet zal men bij het bouwen van een schakeling doorgaans eerst in de rommelbak kijken wat er zoal aan IC's in ligt en vervolgens de gewenste functies met de gevonden buit trachten te verwezenlijken. Daarom worden ook de mogelijkheden besproken om met bepaalde basisbouwstenen andere basisfuncties samen te stellen. Hierbij komt Demorgan's theorema aan de orde, alsook de mogelijkheid om functies in NAND- of NOR-techniek te maken. In de industrie wordt veel gebruik gemaakt van deze varianten, omdat zij voordelig zijn. Nog andere theorieen om tot een zo klein mogelijke schakeling te komen om een gewenste

6.1 Inleiding

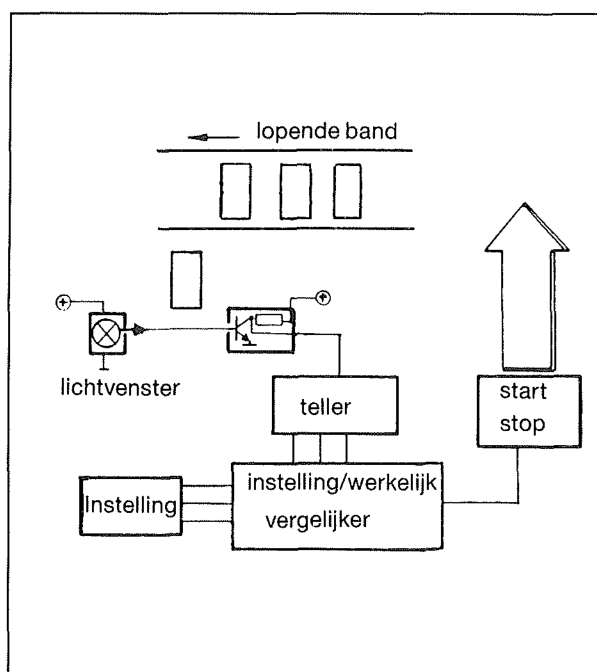
functie te komen zullen aan de orde komen.

Door poorten weer te combineren kunnen a-, bi- en monostabiele multivibratoren worden gerealiseerd. Deze samenstellingen worden weer als een uitgebreidere basisfunctie gezien. De meer gebruikelijke naam is 'flip-flop'. Bij bespreking van deze onderdelen is het echt raadzaam om de voorbeelden na te bouwen. Dit zal de begrijpelijkheid van de uitleg zeer ten goede komen.
In figuur 3/6.1-4 zien we een toepassing van een 'set/reset flip-flop als contactdender-onderdrukker.
Flip-flops vormen op hun beurt weer de bouwstenen voor deler- en teller schakelingen. De huidige digitale technieken maken op een zo uitgebreide manier gebruik van deze schakelingen, dat ze ook weer een basisfunctie vormen. Denk aan timers, analoog naar digitaal omzetters etc. Zowel vooruit- als achteruit tellende tellers worden besproken, figuur 3/6.1-5. Combinaties van tellers kunnen gecodeerde uitgangsstanden leveren, die weer vergeleken kunnen worden met ingestelde codes en dan kunnen worden gebruikt om beslissingen als 'gelijk', 'groter dan', 'kleiner dan' e.d. te nemen. Met behulp van decodeerschakelingen kunnen tellerstanden weer worden omgezet naar codes, die via opto-electronische componenten (displays) leesbare resultaten opleveren. Ook deze complexere componenten komen ter sprake en de methode om hun complexiteit tot begrijpelijke basisfuncties terug te brengen wordt besproken.

Flip-flops kunnen ook worden gebruikt om schuifregisters te maken. Schuifregisters vindt men in schakelingen als lichtorgels. Aansluitend op alle eerder genoemde principes en theorieen zullen de principes en functie van analoog/digitaal omzetter en digitaal/analoog omzetters worden behandeld. In de figuren 3/6.1-6 en 7 kunt u nog wat voorbeelden zien van gebruik van verschillende digitale bouwstenen tot een totaalschakeling.

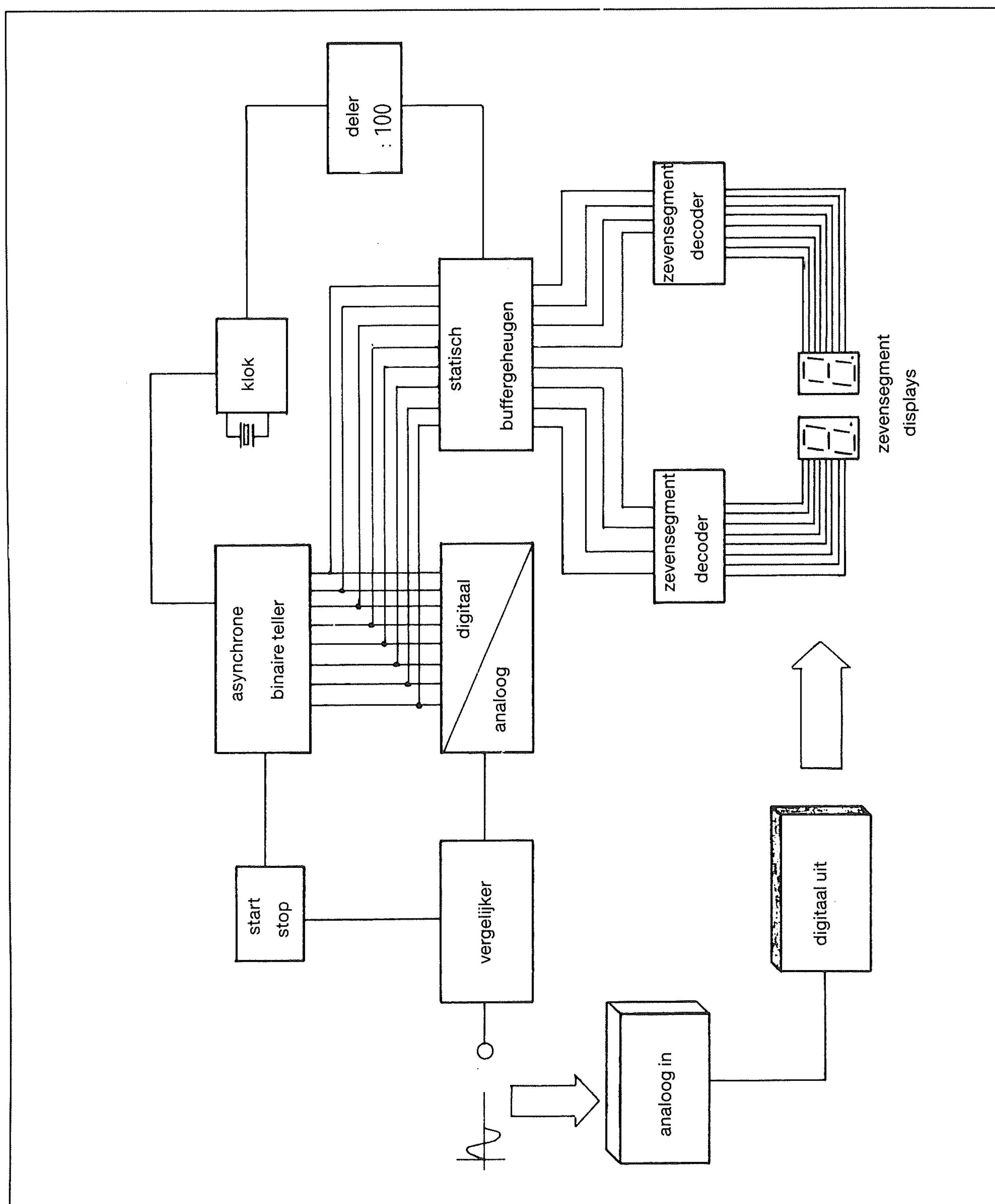


Figuur 3/6.1-4: Schakeling voor onderdrukking van contactdender.



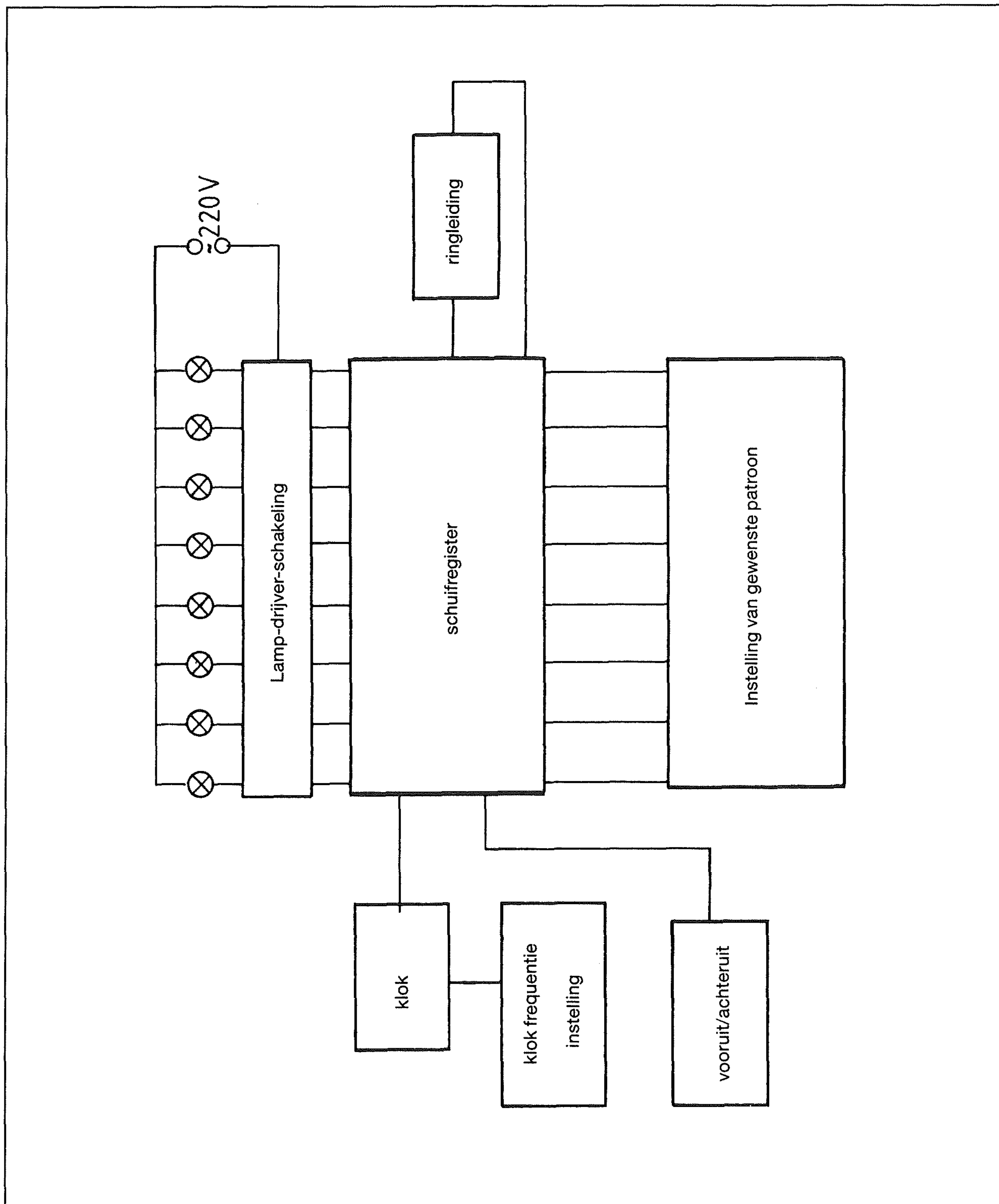
Figuur 3/6.1-5: Verpakken van steeds dezelfde hoeveelheid van produkten.

6.1 Inleiding



Figuur 3/6.1-6: In een analoog/digitaal omzetter, worden veel basisfuncties gebruikt.

6.1 Inleiding



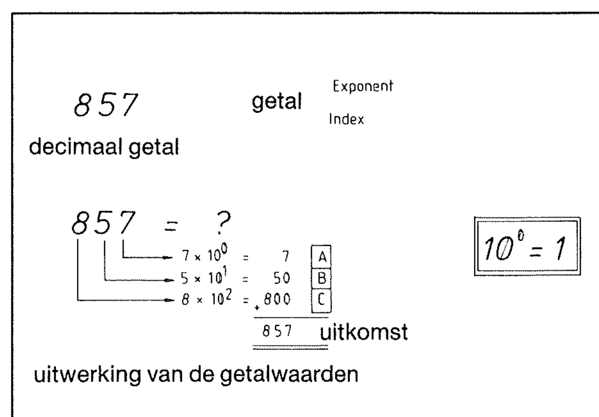
Figuur 3/6.1-7: Een gedetailleerde ontwikkelingsbeschrijving van dit 8-kanaals looplicht met talrijke mogelijkheden komt in een latere aanvulling aan de orde.

3/6.2

De getalstelsels

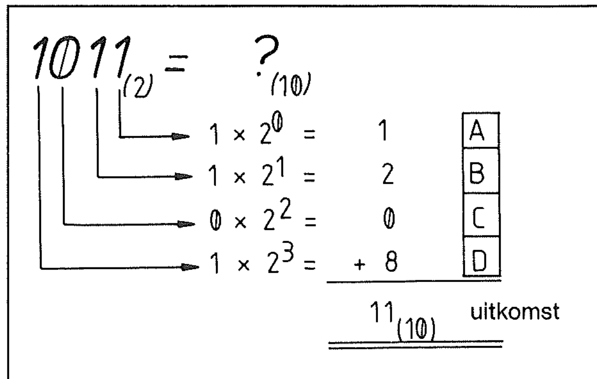
In het binaire stelsel wordt met de cijfers 0 en 1 gerekend, terwijl in het decimale stelsel met de cijfers 0 tot 9 wordt gewerkt. In de exponentiele schrijfwijze is 10 het grondtal van het decimale stelsel. Als voorbeeld zien we in fig. 3/6.2-1 hoe bijvoorbeeld het getal 857 in machten van 10 wordt uiteengerafeld. Als men de afzonderlijke uitkomsten A tot C bekijkt, dan ziet men, dat als men de posities der cijfers van rechts naar links nummert te beginnen met 0, elk cijfer wordt vermenigvuldigd met 10 tot de macht x, waarbij x de positie is. Optelling van de uitkomsten geeft de uiteindelijke waarde. Hetzelfde principe kan men toepassen op het binaire stelsel, met als enige verschil dat het grondtal niet 10 is maar 2. In fig. 3/6.2-2 ziet u hoe dit wordt gedaan en hoe het binaire getal 1011 overeenkomt met het decimale getal 11. De indexen 2 en 10 geven deze grondtallen aan. Dit zijn de getallen, die tussen haakjes onderaan de getallen zijn geplaatst. (zie ook tabel 3/6.2-1). Aangezien in het binaire stelsel wat grotere getallen veel digits hebben en daardoor nogal onleesbaar worden zijn er andere stelsels ingevoerd, die zich gemakkelijk in het binaire stelsel laten omzetten, maar die veel beter leesbaar zijn. Deze stelsels zijn het octale stelsel met grondtal 8 en het hexadecimale stelsel met grondetal 16. Om de afzonderlijke waarden alle eenduidig aan te geven zijn in het hexadecimale stelsel naast de cijfers 0 tot en met 9 nog de letters A tot en met F in gebruik om de resterende 5 waarden aan te duiden. (A=10, B=11, C=12, D=13, E=14 en F=15).

Het is interessant te onderzoeken wat het grootste voor te stellen getal is, dat bij een gegeven aantal cijfers (digits) in het gebruikte getalstelsel is weer te geven. Om dit te verduidelijken een voorbeeld: Wat is het grootste getal dat met 5 digits in het octale stelsel kan worden voorgesteld.
Natuurlijk kan men de methode uit fig. 3/6.2-2 toepassen. Dit is echter nogal bewerkelijk.



Figuur 3/6.2-1: Uiteenrafelen van een decimaal getal in de exponentiële schrijfwijze.

6.2 De getalstelsels



Figuur 3/6.2-2: Omrekening van een binair getal in een decimaal getal.

stelsel	grondtal (index)	positie waarde	gebruikte cijfer aanduidingen
decimaal	10^{*}	10^3 10^2 10^1 10^0 1000 100 10 1	0,1,2,3,4,5,6,7,8,9
binair	2	2^3 2^2 2^1 2^0 8 4 2 1	0,1
octaal	8	8^3 8^2 8^1 8^0 512 64 8 1	0,1,2,3,4,5,6,7
hexa decimaal	16	16^3 16^2 16^1 16^0 4096 256 16 1	0,1,2,3,4,5,6,7,8,9, A,B,C,D,E,F
polynomisch (algemeen geldend)	B	B^n B^2 B^1 B^0	0, ,B-1

Tabel 3/6.2-1: Overzicht van de getalstelsels.

Fig. 3/6.2-3 laat zien op welke wijze dit getal snel kan worden berekend uit de formule. Voor onze vraagstelling luidt het antwoord dus: $(8^5 - 1) = 32767$. Het antwoord is in het decimale stelsel. Bij rekenen in andere stelsels zijn voortdurend omrekeningen noodzakelijk. Tabel 3/6.2-2 geeft alle mogelijke omrekenings wijzen.

$$B^n - 1$$

waarbij B = grondtal
n = aantal digits

Figuur 3/6.2-3: Formule voor het grootst weer te geven getal.

Rekenen in getallenstelsels.

Natuurlijk zijn berekeningen, zoals optellen, aftrekken, vermenigvuldigen en delen ook in andere getalstelsels dan het decimale mogelijk. Daartoe moeten we echter eerst deze functies, die we eigenlijk zonder aan de regels te denken uitvoeren eens analyseren. Daarbij komen de gevolgde regels weer naar boven. Als we die maar consequent in de andere getalstelsels toepassen komt het antwoord vanzelf uit de bus.

```
   7
   5
 + 1      onthouden
 ----
  12      antwoord
```

Figuur 3/6.2-4: De bewerking 'optellen' in het decimale stelsel.

Laten we eerst eens een optelling maken. Daartoe bekijken we fig. 3/6.2-4. Opgave: tel 7 en 5 op. Het antwoord is hoger dan het grondtal van het stelsel waarin we rekenen (10). We schrijven daarom de laatste digit (= cijfer) van de uitkomst op, de rest onthouden we voor optelling bij de eerst hogere digit van het getal, of zo u wilt voor optelling bij de volgende positie. De optelling is gereed en het antwoord bekend als alle posities zijn opgeteld. Een optelling in het hexadecimale stelsel (grondtal=16) is te zien in fig. 3/6.2-5. De optelling van de cijfers 'F' ($=15_{[10]}$) en 'A' ($=10_{[10]}$) gaat over het grondtal 16. (15+10=25). We dragen dus 1 x het grondtal over naar de volgende positie en schrijven de rest ($25_{[10]}-16_{[10]}=9_{[10]}=9_{[16]}$) op in het antwoord. Daarna tellen we de vol-

6.2 De getalstelsels

gende positie op gelijke wijze op.

$1F_{(16)}$
$1A_{(16)}$
+ 1 onthouden
$39_{(16)}$ antwoord

Figuur 3/6.2-5: Hexadecimaal optellen

$1011_{(2)}$
$101_{(2)}$
+ 1 1 1 1 onthouden
$10000_{(2)}$ antwoord

Figuur 3/6.2-6: Binair optellen

Bij het binaire systeem blijven dezelfde regels van kracht. Men moet er slechts op letten, dat het grondtal 2 is. Zodra dus het resultaat 2 bedraagt, moet men 0 opschrijven en 1 onthouden voor de volgende positie.

Binair aftrekken

Dit vereist iets meer nadenken, dat wil zeggen, dat aangezien rekenen in een ander getalstelsel niet routinematig wordt gedaan men even moet opletten. Ook hier is door veel doen echter dezelfde routine op te bouwen, die we in al onze schooljaren hebben opgebouwd in het decimale stelsel. Als men van het decimale systeem uitgaat, kan men de regels weer analiseren en toepassen op de andere systemen. Zie fig. 3/6.2-7: Opgave: trek 749 af van 842. Aangezien van 2 geen 9 kan worden afgetrokken, lenen we 1 vol grondtal (10) van de volgende positie. Van de som van 2 en 10 = 12 kan 9 wel worden afgetrokken. De aftrekking is gereed, als de positie met de hoogste waarde is behandeld, daarbij de geleende eenheden meegerekend. Als we het principe op het binaire systeem toepassen krijgen we het plaatje van figuur 3/6.2-8. Het belangrijkst om te onthouden is in dit geval, dat een geleende eenheid van een hogere positie de waarde van het grondtal heeft in de daarop volgende lagere positie. Ingeval van de berekening van figuur 3/6.2-8 wordt er 1 grondtal geleend bij de derde positie (van links). Dit levert dus 2 op in de tweede positie. Hiervan lenen we er weer 1, die in de laagste positie weer 2 is. 2−1=1 en daarmee is het resultaat van de laagste positie bekend. De laagste (=rechtse) positie van een getal wordt dikwijls in computerterminilogie aangeduid met LSB (Least Significant Bit). De hoogste positie wordt overeenkomstig aangeduid met MSB (Most significant Bit).

842 aftrektal
749 aftrekker
− 1 1 geleend
93 antwoord

Figuur 3/6.2-7: Decimaal aftrekken

6.2 De getalstelsels

omrekenings-tabel

gegeven getalstelsel	gevraagde getalstelsel	Bewerking (polynomisch)	voorbeeld
willekeurig	decimaal	$XX_{(B)} = ?_{(10)}$ → $X \times B^0 = Y$ → $X \times B^1 = + W$ Antwoord $? = Z$	$27_{(8)} = ?_{(10)}$ → $7 \times 8^0 = 7$ → $2 \times 8^1 = + 16$ 23
decimaal	willekeurig	$XXX_{(10)} = ?_{(B)}$ → : B = E1 rest R1 → E1 : B = E2 rest R2 → E2 : B = E3 rest R3 → E3 : B = 0 rest R4 Antwoord ? = $R4\,R3\,R2\,R1_{(B)}$	$712 = ?_{(16)}$ 712 : 16 = 44 rest 8 44 : 16 = 2 rest 12 = C 2 : 16 = 0 rest 2 $? = 2C8_{(16)}$
binair	octaal	$X5\,X4\,X3\,X2\,X1_{(2)} = ?_{(8)}$ Deling in drie groepen van links naar rechts, alsook aanvullen met 0 0 X5 X4 X3 X2 X1 vergelijking met de tabel levert VI VII Antwoord ? = $VI\,VII_{(8)}$	$10101_{(2)} = ?_{(8)}$ 010 101 2 5 $? = 25_{(8)}$

octale waarden tabel

X3	X2	X1	V
0	0	0	0
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4
1	0	1	5
1	1	0	6
1	1	1	7

gegeven getalstelsel	gevraagde getalstelsel	Bewerking (polynomisch)	voorbeeld
octaal	binair	$VI\,VII_{(8)} = ?_{(2)}$ VII = X3 X2 X1 VI = ~~0~~ X5 X4 vervalt Antwoord ? = $X5\,X4\,X3\,X2\,X1_{(2)}$	$113_{(8)} = ?_{(2)}$ VI = 011 VII = 001 VIII = ~~00~~1 $? = 1001011_{(2)}$

X = cijferwaarde
B = grondtal

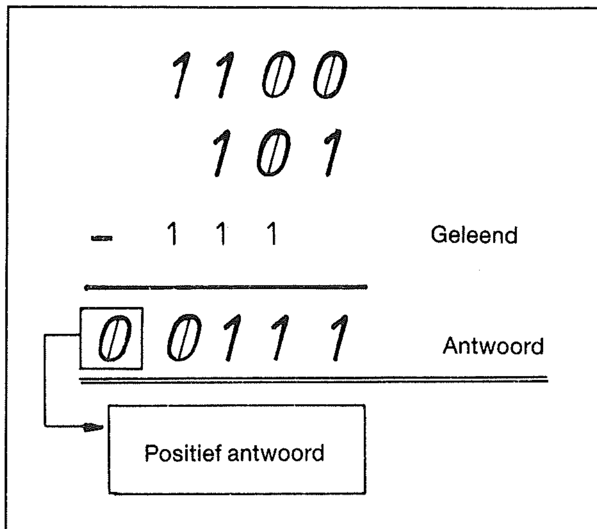
Tabel 3/6.2-2: deel 1.

6.2 De getalstelsels

gegeven getalstelsel	gevraagde getalstelsel	Bewerking (polynomisch)	voorbeeld
binair	hexadecimaal	[X7 X6 X5 X4 X3 X2 X1] (2) = ? (16) deling in groepen van vier van rechts naar links alsook aanvullen met 0 [0 X7 X6 X5] [X4 X3 X2 X1] vergelijking met de tabel levert [VI] [VII] antwoord ? = [VI VII] (16)	$1011101_{(2)} = ?_{(16)}$ 0101 1101 5 D $? = 5D_{(16)}$
		X4 X3 X2 X1 | V 0 0 0 0 | 0 0 0 0 1 | 1 0 0 1 0 | 2 0 0 1 1 | 3 0 1 0 0 | 4 0 1 0 1 | 5 0 1 1 0 | 6 0 1 1 1 | 7 1 0 0 0 | 8 1 0 0 1 | 9 1 0 1 0 | A 1 0 1 1 | B 1 1 0 0 | C 1 1 0 1 | D 1 1 1 0 | E 1 1 1 1 | F	Hexadecimale waarden tabel
hexadecimaal	binair	[VI VII] (16) = ? (2) = [X4 X3 X2 X1] = [X8 X7 X6 X5] (X8 X7 doorgestreept) antwoord : ? = [X6 X5 X4 X3 X2 X1] (2)	$2E3_{(16)} = ?_{(2)}$ = 0011 = 1110 = 0010 antwoord ? = $10\ 1110\ 0011_{(2)}$
hexadecimaal	octaal	[VI VII] (16) = ? (8) binaire notatie = [X4 X3 X2 X1] = [X8 X7 X6 X5] deling in groepen van drie [0 X8 X7] [X6 X5 X4] [X3 X2 X1] [VIII] [VVI] [VV] antwoord ? = [VIII VVI VV] (8)	$7B_{(16)} = ?_{(8)}$ = 1011 = 0111 001 111 011 1 7 3 $? = 173_{(8)}$
octaal	hexadecimaal	op gelijke wijze: eerst octaal – hexadecimaal dan hexadecimaal – octaal	

Tabel 3/6.2-2: deel 2

6.2 De getalstelsels

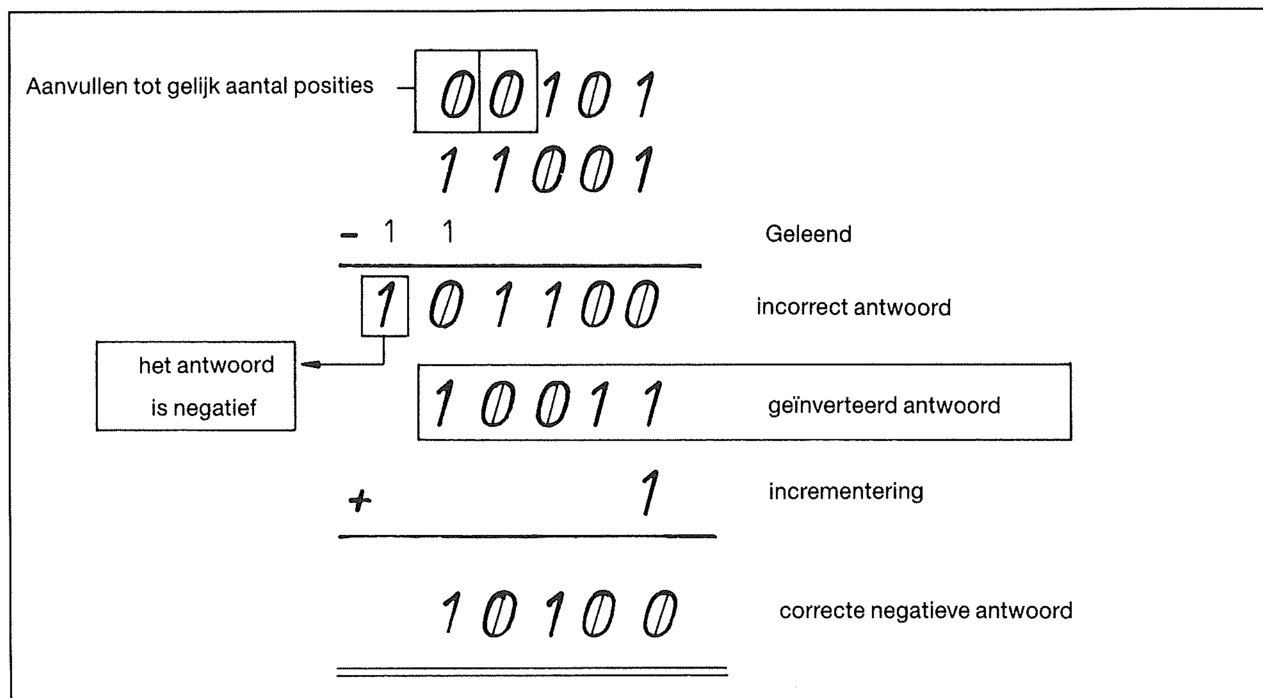


Figuur 3/6.2-8: Binair aftrekken

Wat gebeurt er als de aftrekker groter is dan het aftrektal? Daarvoor is het voorbeeld van fig. 3/6.2-9. Zoals te zien is heeft het aftrektal minder digits dan de aftrekker en is dus zeker kleiner. We beginnen met de aftrekker aan te vullen met nullen, totdat aftrekker en aftrektal een gelijk aantal digits heeft. Daarmee verandert de waarde van het aftrektal niet. Als men nu gaat aftrekken, komt men tot de ontdekking, dat voor de bewerking van de meest linkse positie (MSB) nog een grondtal zou moeten lenen. We stellen voor alsof er een volgend digit was, maken de aftreksom af en schrijven het resultaat in de zesde positie. Dit fictieve volgende digit wordt het 'carry-bit' of kortweg 'carry' genoemd. Het carry-bit bepaald dus of het getal positief is of negatief. (In computers is dit carry-bit aanwezig als een extra bit van het rekenregister.) Door deze truc lijkt het getal positief. We weten echter dat dit niet correct is omdat we een grondtal uit het niets tevoorschijn hebben getoverd.
In de binaire rekenwijze hebben we een regel ingesteld, die zegt, dat ingeval het MSB 1 is, daarmee wordt aangegeven, dat het getal negatief is. Aangezien computers echter niet met negatieve getallen werken moeten we dit incorrecte antwoord omzetten in een voor de computer positief getal, waaraan we toch kunnen zien dat het getal negatief is. Daartoe verdeeld men in een computer het grootst voor te stellen getal (afhankelijk van het aantal bits, weet u nog) in twee helften.
Zolang het MSB 0 is noemen we het getal positief, als het MSB 1 wordt noemen we het getal negatief. Voorbeeld: als de computer met 4 bits werk, is het grootst voor te stellen getal $1111_{[2]}=15_{[10]}$. 0000 . . . 0111 is positief, 1000. . . 1111 is negatief. We noemen deze rekenwijze de 'two's complement' rekenwijze.
Om nu het verkregen resultaat van de uitgevoerde berekening weer terug te brengen naar 5 digits inverteren we eerst het incorrecte antwoord, dwz., dat alle nullen enen worden en omgekeerd. Het resultaat is $10011_{[2]}=19_{[10]}$ Dit is echter nog steeds incorrect. De computer 'weet' dit en telt bij het resultaat 1 op (absoluut gezien). Dit geeft als eindresultaat $10100_{[2]}=20_{[10]}$. Dit is het juiste antwoord, van $101_{[2]}=5_{[10]}$ min $11001_{[2]=25[10]}$. De computer bekijkt dus waarde van de eerst hogere positie na de aftrekking en besluit op grond van die waarde of het antwoord al dan niet negatief is.

Wat we in wezen gedaan hebben is van aftrekken optellen gemaakt met gebruikmaking van 'two's complement' rekening. Probeert uzelf maar eens. Aftrekken is hetzelfde als het 'two's complement' van de aftrekker bij het aftrektal optellen.

6.2 De getalstelsels



Figuur 3/6.2-9: De notatie van negatieve binaire getallen

Hiermee behoeven we dus de computers niet te voorzien van gecompliceerde logica om af te kunnen trekken.

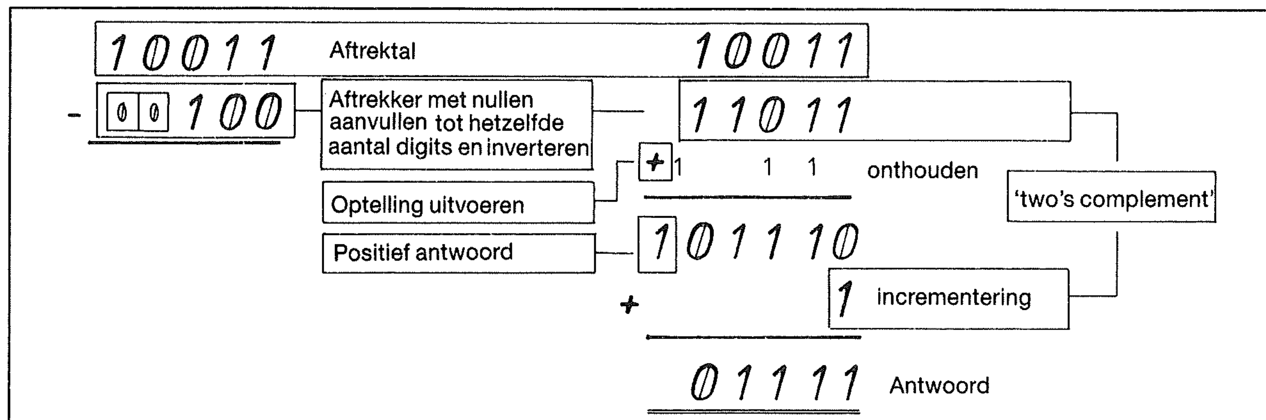
Binair vermenigvuldigen.

Vermenigvuldigen is van van alle bewerkingen in het binaire stelsel verreweg de eenvoudigste. Het gaat op exact dezelfde manier als in het decimale stelsel. Alleen hoeft u nu nog slechts de tafel van 0 tot 1 te kennen. Met het optellen van de resultaten is het produkt bekend. Zie figuur 3/6.2-11. Rekenen met andere getalstelsels dan binair en decimaal komt in de praktijk zelden voor.

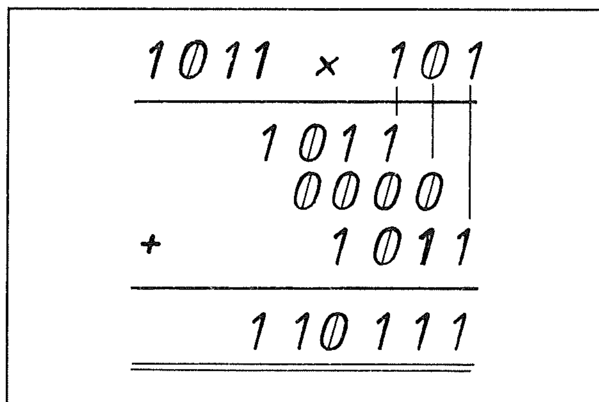
Binair delen

Delen is eigenlijk onderzoeken, hoe vaak de deler in het deeltal voorkomt. Men kan dus de deler van het deeltal aftrekken en tellen hoeveel gehele malen dat mogelijk is. Eenvoudige computers passen deze wijze toe. Het kost echter nogal wat tijd. Derhalve kiest men voor hetzelfde systeem van staartdelingen als in het decimale stelsel, zie fig. 3/6.2-12.
Eerst neemt men het 'two's complement van de deler en telt dit bij het deeltal op. Als het 'carry-bit' 1 is, dan staat het resultaat vast. Als het carrybit' 0 is, dan moet een correctie worden uitgevoerd met de niet gecomplementeerde deler, waarbij het 'carry-bit' niet meer van belang is. Na de eerste deling wordt het digit van de volgende positie bij de rest genomen en herhaalt men de procedure. De deling is af, wanneer aan het eind nullen resteren, of de deling na de komma wordt afgebroken.

6.2 De getalstelsels

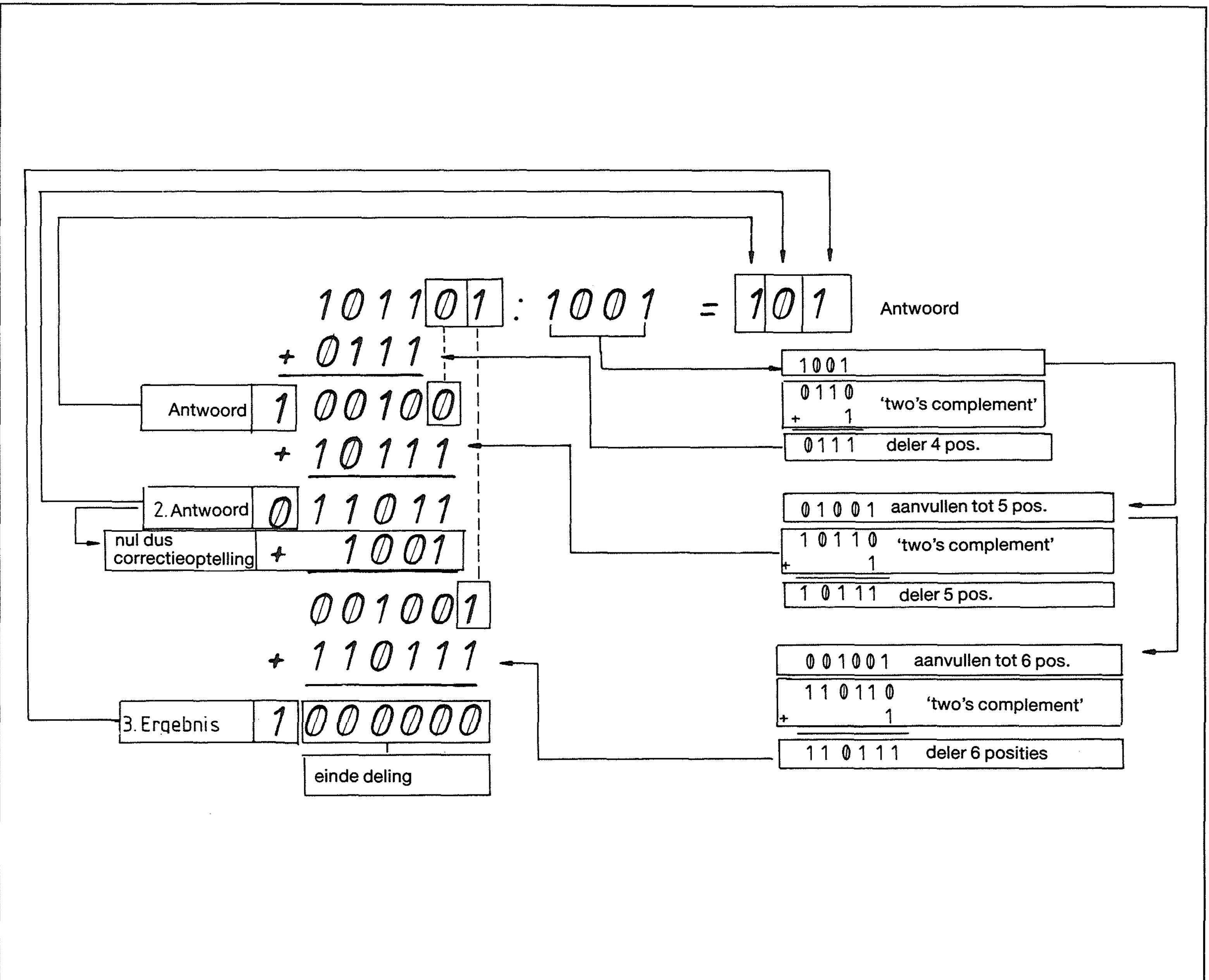


Figuur 3/6.2-10: Aftrekken door optellen met 'two's complement' rekenwijze



Figuur 3/6.2-11: Binair vermenigvuldigen

6.2 De getalstelsels



Figuur 3/6.2-12: Binair delen zonder rest

6.2 De getalstelsels

3/6.3

Schakelingen van digitale functies

Inhoud

3/6.3.1 De basisfuncties
(verschenen in de 12e aanvulling)
3/6.3.1.1 De inverter
3/6.3.1.2 De AND-poort
3/6.3.1.3 De OR-poort

3/6.3.2 De afgeleide functies
(verschenen in de 12e aanvulling)
3/6.3.2.1 De NAND-poort
3/6.3.2.2 De NOR-poort
3/6.3.2.3 Enable/Inhibit
3/6.3.2.4 EXCLUSIV-OR-poorten
3/6.3.2.5 EXCLUSIV-NOR-poorten

6.3 Schakelingen van digitale functies

Elke elektronische schakeling heeft als doel van een ingangssignaal of een combinatie van ingangssignalen een uitgangssignaal te maken. Zo maakt een versterker van een klein ingangssignaal een groot uitgangssignaal. In de digitale techniek, kunnen zowel ingangssignalen als uitgangssignalen slechts twee gedefinieerde toestanden aannemen. Een digitale schakeling analyseert de aan de ingang aangeboden spanning, besluit welke van de twee gedefinieerde toestanden door deze spanning wordt vertegenwoordigd en vormt uit een of meerdere ingangssignalen uiteindelijk een of meerdere uitgangssignalen. Een voorbeeld: Een digitale schakeling controleert de toegang tot een laboratorium. De deur wordt slechts dan ontgrendeld als de aangeboden code correct is, een geldige toegangskaart in de kaartlezer is gelegd en het tijdstip correct is. Voor elke vervulde conditie kan de schakeling op zijn uitgang een hoog of laag niveau geven (welk is afhankelijk van de schakeling en doet voor het principe niet ter zake). Als aan alle ingangscondities gelijktijdig is voldaan, levert de schakeling een uitgangssignaal, dat kan worden gebruikt om een openingsmechanisme van de deur in werking te stellen. In de digitale techniek bereikt men dit resultaat, door verschillende logische functies met elkaar te combineren. Dat wil zeggen, dat schakelingen met bepaalde (basis-)functies met elkaar worden verbonden tot een totale schakeling die uiteindelijk de gewenste functie oplevert. Veel voorkomende combinaties van met elkaar verbonden logische functies worden door de IC-fabrikanten als kant en klare logische bouwstenen aangeboden. In de loop der tijd heeft de miniaturisering en de voortschreiding der techniek er voor gezorgd, dat er schakelingen op de markt zijn, die uit miljoenen elementen bestaan. Afhankelijk van de complexiteit van de schakelingen spreekt men van SSI (= small scale integration), LSI (= large scale integration), VLSI (= very large scale integration). De microprocessor is een goed voorbeeld van de laatste categorie. Het eind is echter nog niet in zicht. Het aantal elementen, dat kan worden geïntegreerd per oppervlakte eenheid neemt nog steeds toe. Tot voor kort was te voorzien, dat er fysiek een eind komt aan deze 'densiteit'. De fysici hebben nu echter methoden ontdekt om integratie toe te passen in de diepte, waardoor schakelingen drie-dimensionaal kunnen worden opgebouwd. Echter laat u zich niet afschrikken. Hoe complex ook, alle digitale schakelingen zijn opgebouwd uit (en te reconstrueren tot) de basis functies, die we hier gaan beschrijven.

3/6.3.1

De basisfuncties

Aanduiding	INVERT		AND			OR			NAND			NOR			EXNOR			EXOR		
	A	Y	A	B	Y	A	B	Y	A	B	Y	A	B	Y	A	B	Y	A	B	Y
	0	1	0	0	0	0	0	0	0	0	1	0	0	1	0	0	1	0	0	0
	1	0	0	1	0	0	1	1	0	1	1	0	1	0	0	1	0	0	1	1
			1	0	0	1	0	1	1	0	1	1	0	0	1	0	0	1	0	1
			1	1	1	1	1	1	1	1	0	1	1	0	1	1	1	1	1	0

Tabel 3/6.3.1-1: functietabel van digitale basisfuncties.

In de praktijk blijkt het handig te zijn om in één (nou ja één?) oogopslag de functie van een poort of een gehele schakeling te kunnen zien. Hiertoe bedient men zich van 'functie-tabellen'. In een tabel somt men alle mogelijke ingangsvariabelen op en schrijft achter elke combinatie de door de schakeling in dat geval gegenereerde uitgangstoestand(en). De functietabellen van de basis functies der digitale techniek vindt u in tabel 1.

De wiskundige Georg Boole stelde reeds in 1854 een andere methode op om de logische functies uit te drukken. Hij zocht uit, welke ingangstoestanden aanwezig dienen te zijn om aan de uitgang een logische 1 te verkrijgen. De wetten en regels van de Booleaanse algebra worden in een volgend hoofdstuk behandeld. In de analoge techniek is het algemeen gebruikelijk de onderdelen en hun functies in schema's aan te geven door symbolen. In het schema zijn tevens alle verbindingen aangegeven. Hetzelfde geldt voor de digitale techniek. In tabel 2 vindt u de symbolen, die in verschillende normen worden gebruikt en de daarbij behorende Booleaanse vergelijkingen.

3/6.3.1.1
De inverter

De eenvoudigste basisfunctie wordt vervult door een inverter poort. Zoals eigenlijk uit de naam al blijkt, inverteert hij het signaal op zijn ingang. In fig. 1 is deze omkering duidelijk te zien op het scherm van een oscilloscoop.

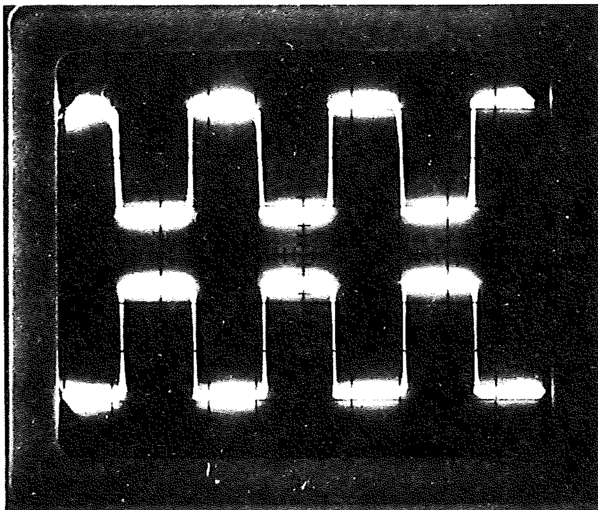
In de TTL serie is een 7406 een voorbeeld van zes inverters in een huis. Het schema ervan vinden we in fig. 2. In de CMOS serie is de 4049 een identiek

6.3 Schakelingen van digitale functies

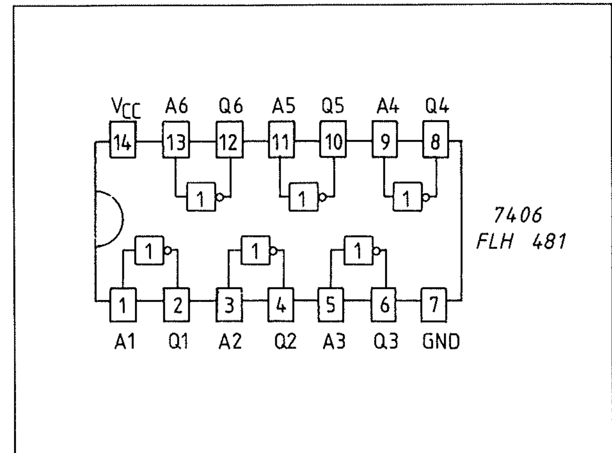
	Functie-omschrijving	DIN 40700	Amerikaanse norm	DIN 40700 nieuw	Booleaanse vergelijking	beschrijving logische functie
basisfuncties	NOT negation inverter	A Y	A Y	A 1 Y	$Y = \bar{A}$	Y=1, als A=0 en omgekeerd
	AND conjunction	A B Y	A B Y	A B & Y	$Y = A \wedge B$	Y=1 als A=1 *en* B=1
	OR disjunction	A B Y	A B Y	A B ≥1 Y	$Y = A \vee B$	Y=1 als A=1 *of* B=1 *of* A=B=1
uitgebreide basisfuncties	NAND	A B Y	A B Y	A B & Y	$Y = \overline{A \wedge B}$	Y=1 als A=0 *of* B=0 *of* A=B=0
	NOR	A B Y	A B Y	A B ≥1 Y	$Y = \overline{A \vee B}$	Y=1 als A=0 *en* B=0
	EXNOR exclusive NOR	A B ≡ Y	A B Y	A B = Y	$Y = (A \wedge B) \vee (\bar{A} \wedge \bar{B})$	Y=1 als (A=1 *en* B=1) *of* (A=0 *en* B=0)
	XOR exclusive OR	A B ≢ Y	A B Y	A B 1 Y	$Y = (\bar{A} \wedge B) \vee (A \wedge \bar{B})$	Y=1 als A # B
	enable	A B Y	A B Y	A B ≥1 Y	$Y = \bar{A} \vee B$	Y=1 als (A=0 *of* B=1) *of* (A=0 *en* B=1)
	enable	A B Y	A B Y	A B ≥1 Y	$Y = A \vee \bar{B}$	Y=1 als (A=0 *of* B=0) *of* (A=1 *en* B=0)
	inhibit	A B Y	A B Y	A B & Y	$Y = \bar{A} \wedge B$	Y=1 als A=0 *en* B=1
	inhibit	A B Y	A B Y	A B & Y	$Y = A \wedge \bar{B}$	Y=1 als A=1 *en* B=0

Tabel 3/6.3.1-2: Symbolen en hun functies.

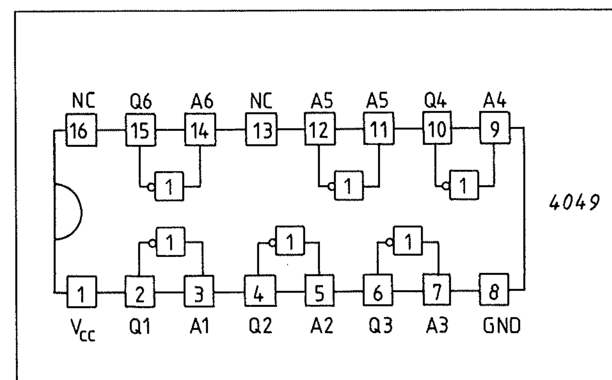
6.3 Schakelingen van digitale functies



Figuur 3/6.3.1.1-1: Oscillogram van een inverter.



Figuur 3/6.3.1.1-2: Zes inverters met open collector.



Figuur 3/6.3.1.1-3: CMOS-inverters.

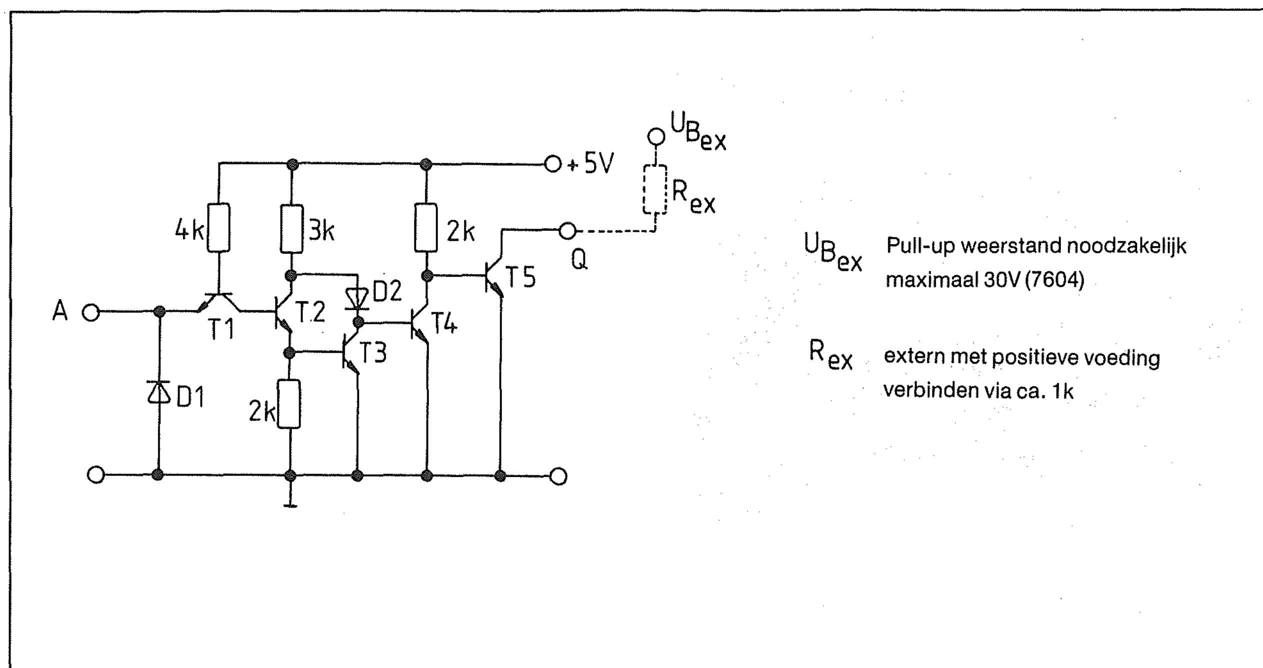
voorbeeld, zij het, dat de behuizing en de aansluiting anders is. Fig. 3 laat dit zien. De invertering wordt bereikt met een omkeertrap. In de TTL techniek wordt deze omkeertrap gerealiseerd met een emittor-basis schakeling, in de CMOS-techniek met een source-schakeling. Zie resp. figuur 4 en 5.

3/6.3.1.2 De AND-poort

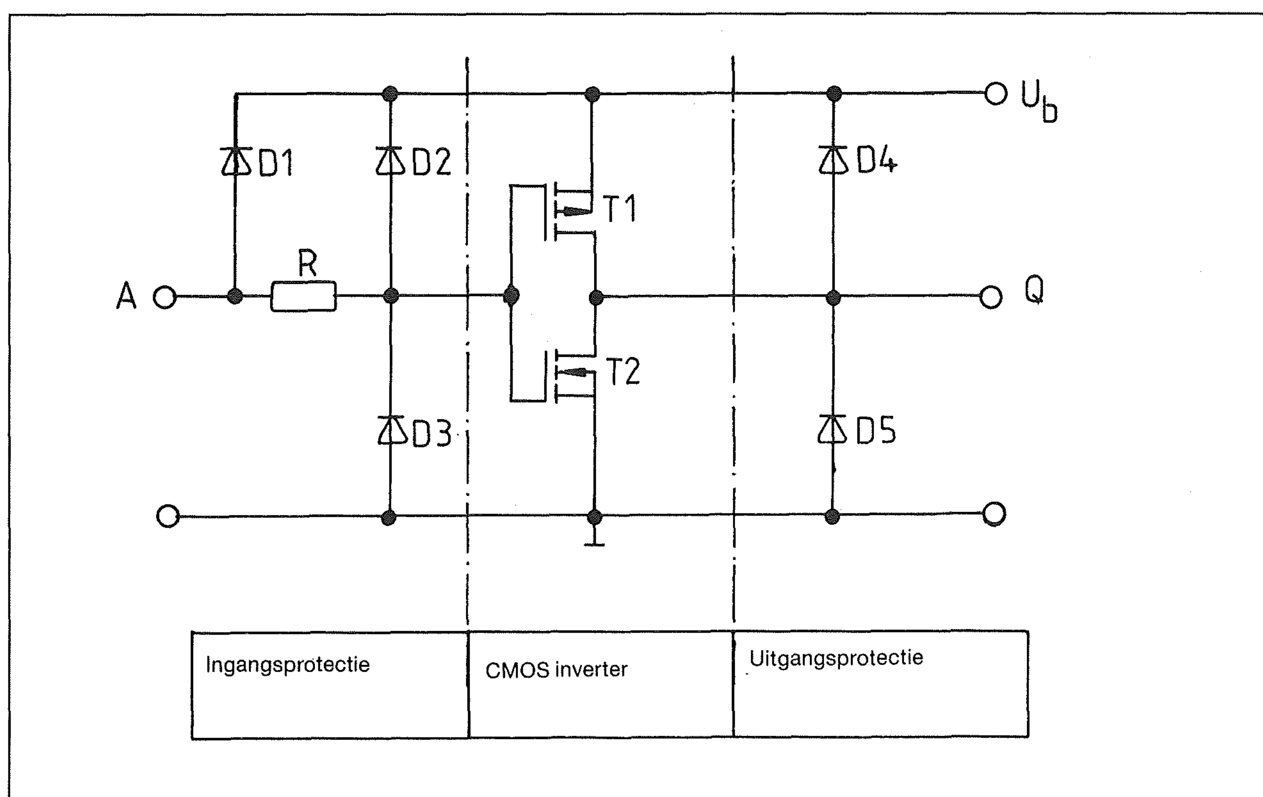
De hiervoor beschreven inverter werkt met slechts een ingangssignaal. De hierna te bespreken basisfuncties zijn gebaseerd op twee of meer ingangssignalen. We beginnen met de AND-poort (een EN-schakeling = conjunctie). Aan de uitgang ontstaat dan slechts dan een logische 1, als alle ingangen tegelijkertijd 1 zijn. De AND-poort laat zich gemakkelijk vergelijken met twee schakelaars, die in serie zijn geschakeld. Het op de bovenkant van de serieschakeling aanwezige hoog niveau verschijnt slechts dan op de uitgang als beide schakelaars gelijktijdig gesloten zijn. Zie figuur 1.

Het TTL IC type 7408 bevat vier AND-poorten met elk twee ingangen. In de TTL techniek wordt een AND-poort gerealiseerd met behulp van een transistor met meer dan een emitter. Zie figuur 2 voor de schematische voorstel van een 7408. TTL-IC's zijn te rangschikken onder stroomgestuurde schakelingen. Teneinde de transistor te laten geleiden zal er stroom door de emitter aansluitingen moeten vloeien. Naast stroomgestuurde schakelingen zijn er de spanningsgestuurde schakelingen. Deze zijn samengesteld uit zelfsperrende P-kanaal of N-kanaal metaal-oxide-halfgeleider-veld-effect transisto-

6.3 Schakelingen van digitale functies

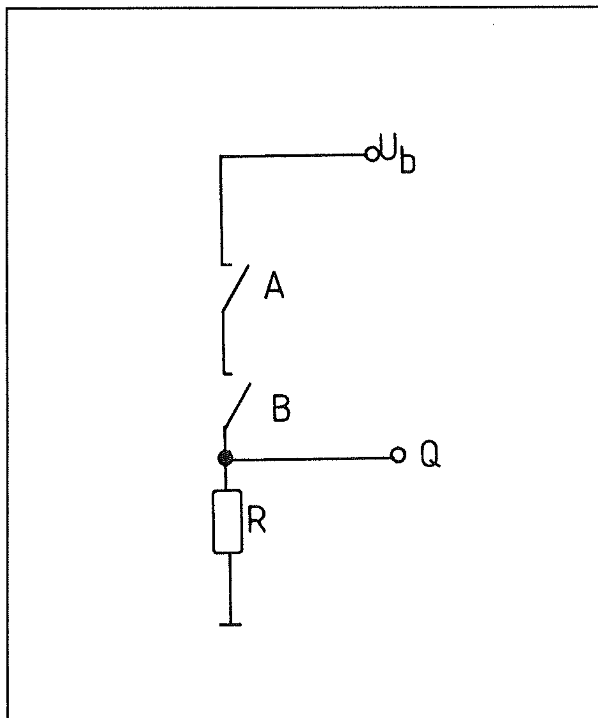


Figuur 3/6.3.1.1-4: Schema 7406 inverter met open collector.

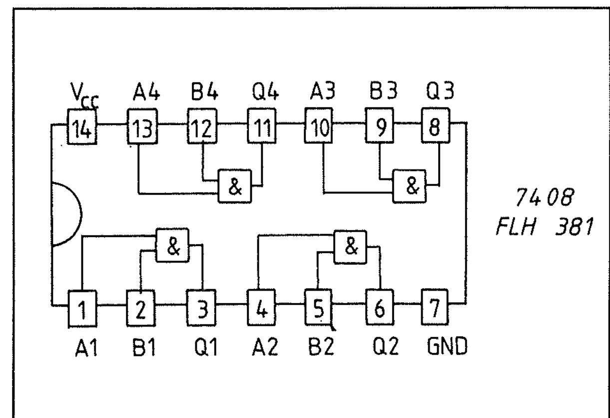


Figuur 3/6.3.1.1-5: Schema 4069 inverter.

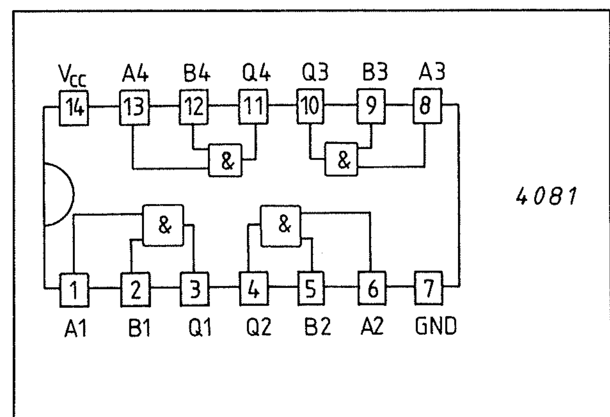
ren (MOSFET). Dit zijn unipolaire transistoren. Het opgenomen vermogen bij deze schakelingen is uiterst gering. Er lopen slechts stromen in de grote orde van nano-amperes (= 10^{-9} Ampere). Het grote nadeel van deze schakelingen is hun gevoeligheid voor statische electriciteit. Daarnaast speelt de snelheid een rol. Door de grotere parasitaire (dit is ongewenste) capaciteit die nu eenmaal een gevolg zijn van de wijze waarop MOSFET's zijn geconstrueerd, zijn deze MOS-typen trager, dan hun TTL broertjes. Een 4081 is een voorbeeld van een CMOS AND-poort. Het IC bevat vier poorten met elk twee ingangen. De schematische weergave vindt U terug in figuur 3. In figuur 4 ziet U links de wijze waarop een TTL AND-poort is samengesteld en rechts zie U de CMOS versie.



Figuur 3/6.3.1.2-1: Het principe van de AND-poort.



Figuur 3/6.3.1.2-2: Vier TTL AND-poorten met elk twee ingangen.

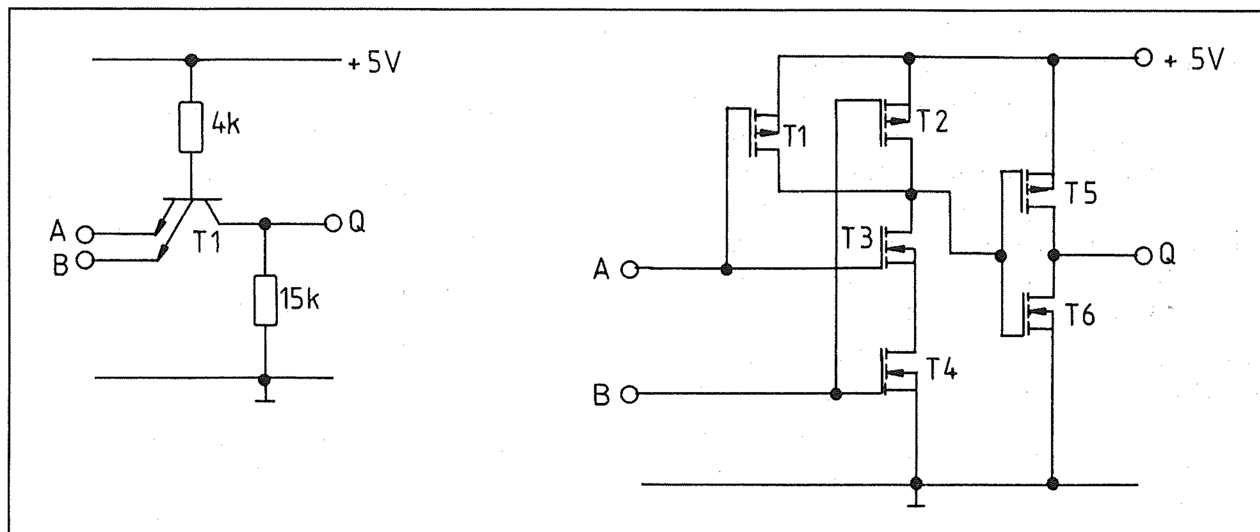


Figuur 3/6.3.1.2-3: Vier CMOS AND-poorten met elk twee ingangen.

3/6.3.1.3
De OR-poort

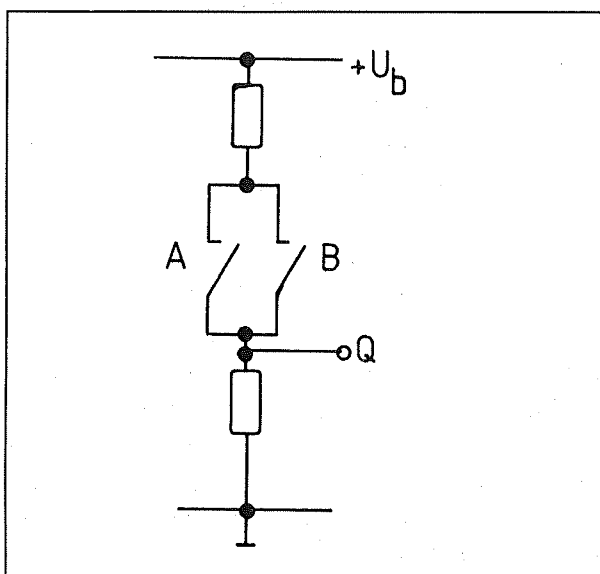
De laatste echte basisfunctie is de OR-poort. (Een OF-schakeling = disjunctie). De uitgang van een OR-poort is altijd logisch 1, wanneer tenminste een van de ingangen 1 is. Maken we weer even een zijstapje naar de schakelaar techniek, dan is de OR-poort voor te stellen als twee parallel geschakelde schakelaars. Zoals U eenvoudig kunt zien in figuur 1 zal de uitgang Q hoog worden zodra een der schakelaars is gesloten. Het TTL IC 7432 is een voorbeeld van OR-poorten. Het bevat vier OR-poorten met elk twee ingangen. In

6.3 Schakelingen van digitale functies



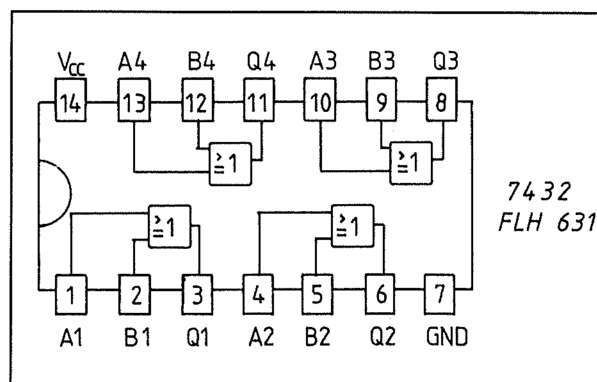
Figuur 3/6.3.1.2-4: De opbouw van een AND-poort in TTL (links) en CMOS (rechts) techniek.

In figuur 2 ziet U de schematische weergave. Hoe zo'n poort is samengesteld vindt U in figuur 3. Als de behoefte bestaat aan een schakeling met een geringer verbruik, dan kan men een CMOS IC toepassen. In figuur 4 vindt U de schematische weergave van een CMOS IC met vier OR poorten, die elk twee ingangen hebben. (Een 4071).



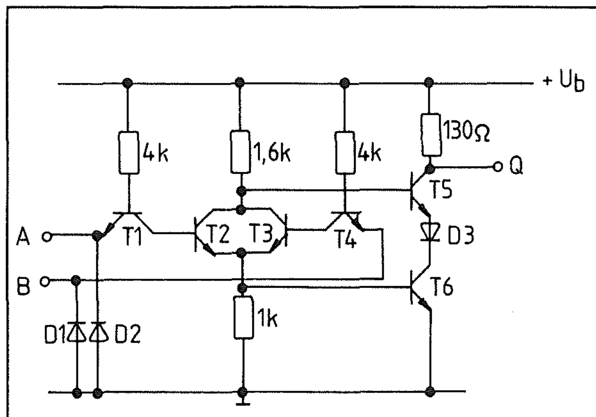
Figuur 3/6.3.1.3-1: Het schakelaar equivalent van een OR-poort.

In figuur 5 ziet U nog een voorbeeld van een CMOS IC (type 4072), dat bestaat uit twee OR-poorten met elk vier ingangen. Tenslotte laat figuur 6 de interne opbouw van een CMOS OR-poort zien.

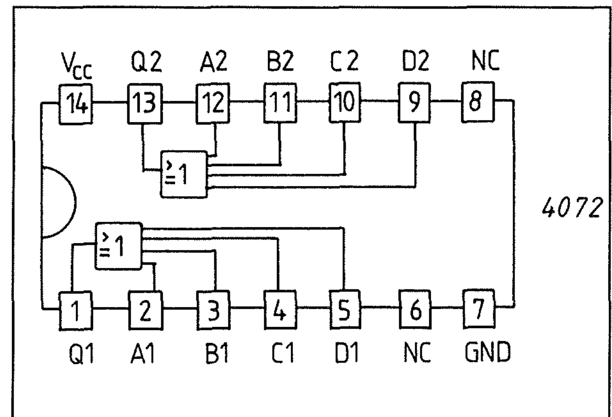


Figuur 3/6.3.1.3-2: Vier TTL OR-poorten met elk twee ingangen.

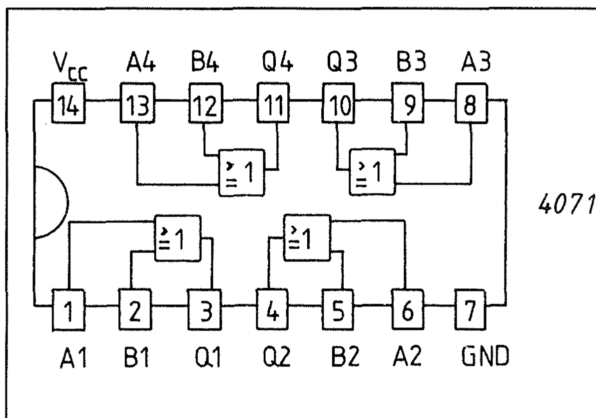
6.3 Schakelingen van digitale functies



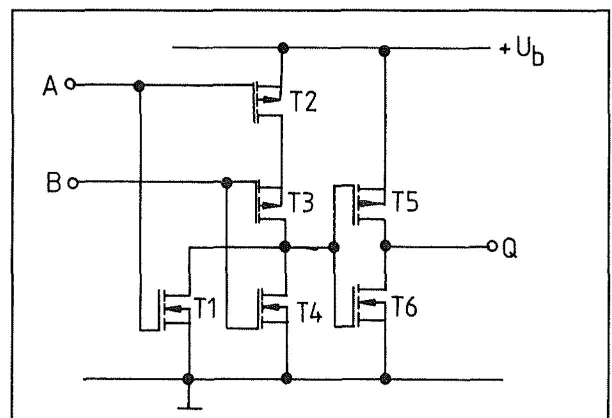
Figuur 3/6.3.1.3-3: TTL OR-poort.



Figuur 3/6.3.1.3-5: Twee CMOS OR-poorten met elk vier ingangen.



Figuur 3/6.3.1.3-4: Vier CMOS OR-poorten met elk twee ingangen.

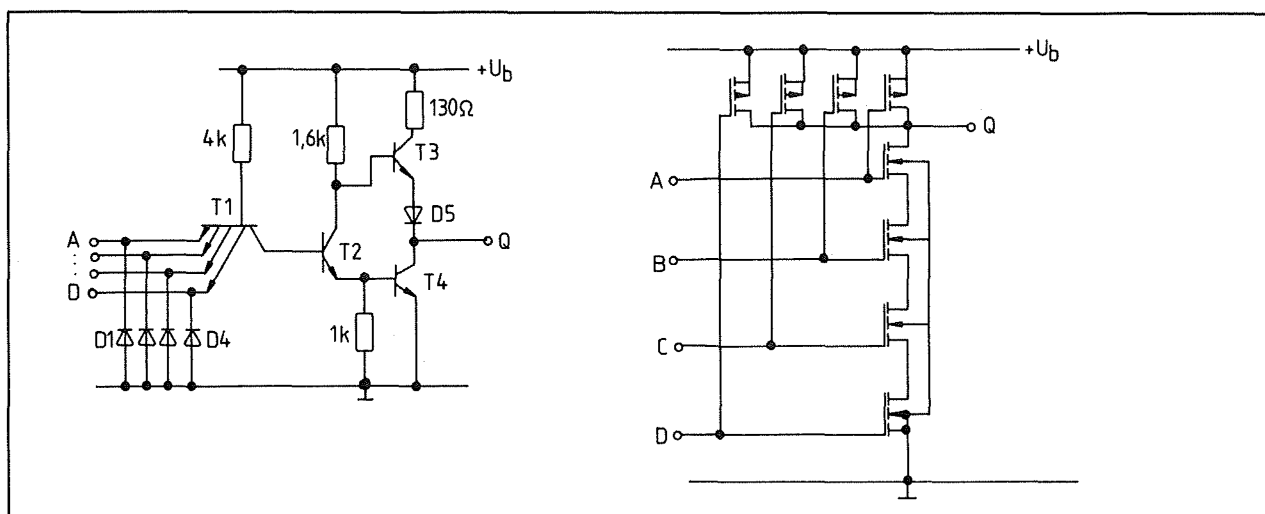


Figuur 3/6.3.1.3-6: CMOS OR-poort.

6.3 Schakelingen van digitale functies

3/6.3.2

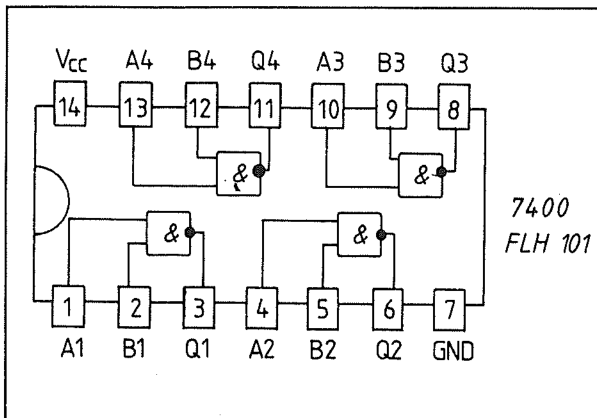
De afgeleide functies



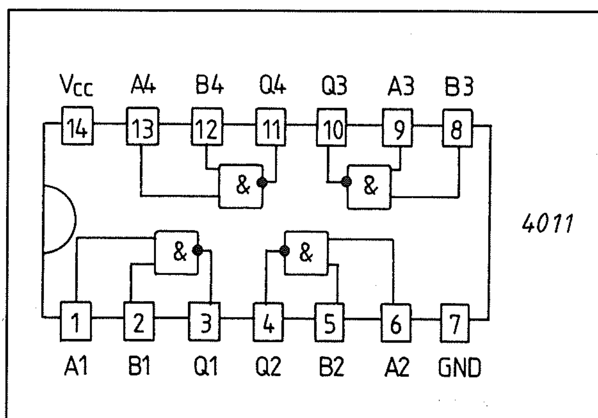
Figuur 3/6.3.2.1-1: TTL resp. CMOS NAND-poort met vier ingangen.

3/6.3.2.1 De NAND-poort

In de afgeleide functies komen de drie hiervoor beschreven basisfunctie terug in combinaties. De NAND-poort kan worden beschouwd als een AND-poort gevolgd door een inverter. In de integratietechniek is de NAND-poort eenvoudiger te realiseren dan een AND-poort. De meeste IC schakelingen zijn dan ook opgebouwd met NAND-poorten. In figuur 1 ziet U links een TTL NAND-poort en rechts de CMOS versie. De NAND-poort heeft op de uitgang een logische 1, als een der ingangen een logische 0 heeft. Waarschijnlijk het meest bekende IC ter wereld is de 7400. Hierin zijn vier TTL NAND-poorten te vinden met elk twee ingangen. De schematische voorstelling vindt U in figuur 2. De overeenkomstige schakeling in de CMOS techniek is een 4011. Zie figuur 3.



Figuur 3/6.3.2.1-2: Vier TTL NAND-poorten met elk twee ingangen.



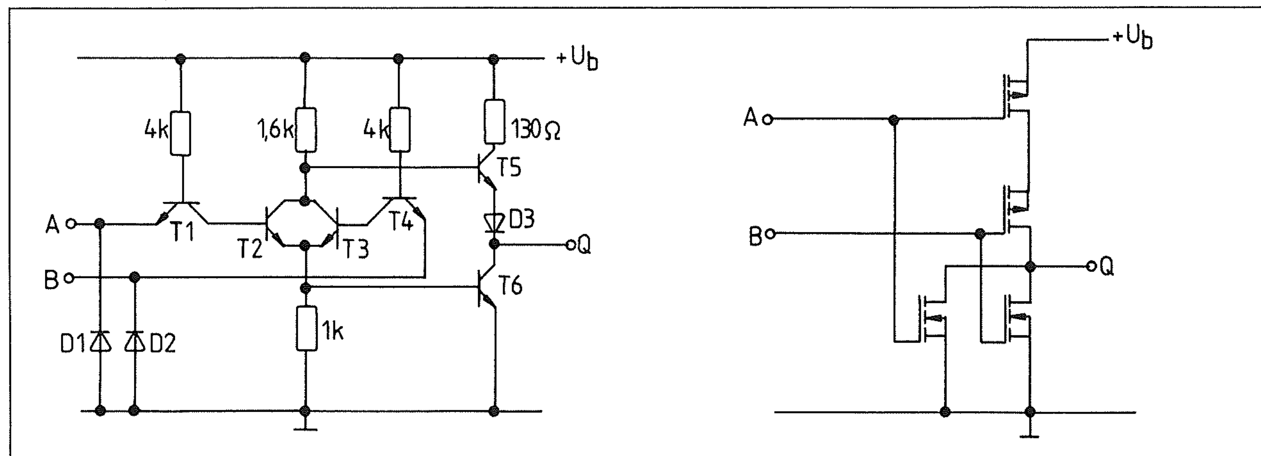
Figuur 3/6.3.2.1-3: Vier CMOS NAND-poorten met elk twee ingangen.

3/6.3.2.2 De NOR-poort

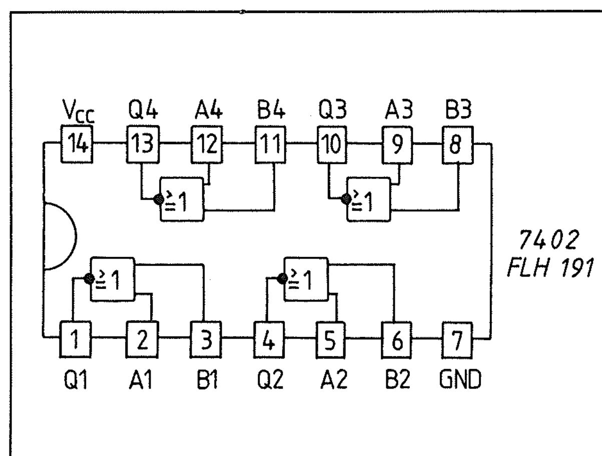
De NOR-poort is zoals U zich waarschijnlijk al kunt voorstellen een combinatie van een OR-poort met achter de uitgang een inverter. In figuur 1 ziet U weer de opbouw van een TTL resp. een CMOS NAND-poort met twee ingangen. De uitgang van een NOR-poort is logisch 1, als beide ingangen als beide ingangen tegelijkertijd logisch 0 zijn. Een 7402 is een TTL NOR-poort. Het IC heeft vier poorten met elk twee ingangen. Voor het aansluitschema zie figuur 2. Natuurlijk is er ook weer een CMOS equivalent, de 4001. U vindt hiervan het aansluitschema in figuur 3. Let op, beide IC's zijn niet pin-compatibel. Als men achter een OR-poort een inverter schakelt, is het resultaat van de totale schakeling een NOR-functie. Evenzo is het mogelijk een NOR-poort weer te inverteren. De resulterende schakeling is weer een OR-poort.

3/6.3.2.3 Enable/Inhibit.

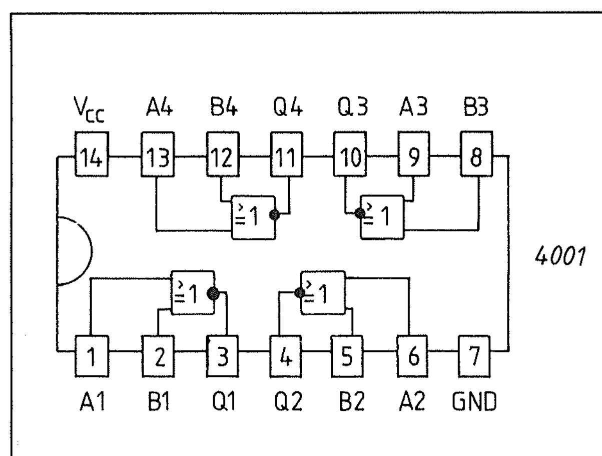
In wezen gaat het bij deze functies om basisfuncties, waarbij een ingang is geïnverteerd. Het is dus zondermeer al duidelijk, dat deze functies met de drie basisfuncties kunnen worden samengesteld. In het geval van een ENABLE-schakeling, is de basisschakeling een OR-poort. Bij de INHIBIT-schakeling is dit een AND-poort. Natuurlijk kan zowel de A- of de B-ingang worden geïnverteerd. In tabel 2 aan het begin van dit hoofdstuk is een en ander in samenhang te zien. Een toepassing van de INHIBIT-functie komt in het volgende hoofdstuk weer terug in de exclusieve OR-poort. De IC fabrikanten hebben geen IC's gemaakt speciaal voor de Enable of Inhibit functies. Deze worden tamelijk wenig gebruikt en zijn ze een keer nodig, dan kan men ze gemakkelijk zelf samenstellen.



Figuur 3/6.3.2.2-1: TTL resp. CMOS NOR-poort met twee ingangen.



Figuur 3/6.3.2.2-2: Vier TTL NOR-poorten met elk twee ingangen.

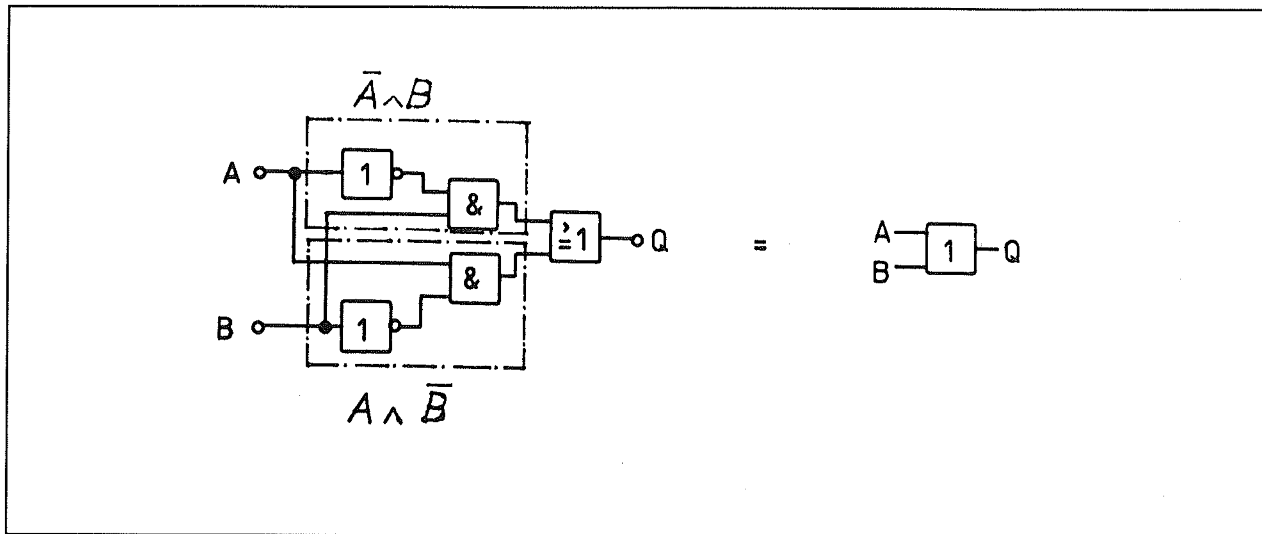


Figuur 3/6.3.2.2-3: Vier CMOS NOR-poorten met elk twee ingangen.

3/6.3.2.4 EXCLUSIV-OR-poorten

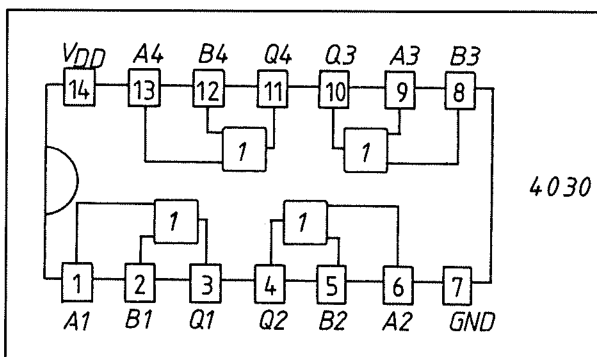
De exclusieve OR-poort wordt meestal kortweg aangeduid met EX-OR-poort of in datahandboeken vaak nog korter met XOR. Kenmerkend aan deze poort is, dat de uitgang slechts dan logisch 1 is, als de logische waarden van de beide ingangen niet gelijk zijn. Een EX-OR-poort kan worden samengesteld uit de basisfuncties. In figuur 1 is dit schematisch weergegeven. Experimenteer zelf maar eens met de basisfuncties. Zoals U ziet zijn voor een EX-OR-poort alle basisfuncties nodig. Omdat de EX-OR-poort toch wel tamelijk veel wordt gebruikt en met basisfuncties opgebouwd de nodige IC's zou kosten ligt het voor de hand, dat de IC fabrikanten een EX-OR-poort als IC leveren. Zowel in de TTL reeks, als in de CMOS reeks komen EX-OR-poorten voor. In figuur 2 ziet U het aansluitschema van een CMOS versie, de 4030. Een belangrijke toepassing van de EX-OR-poorten is het gebruik als een omschakelbare inverter. Beschouwt U ingang A als data ingang en ingang B als de omschakel-commando ingang.

6.3 Schakelingen van digitale functies



Figuur 3/6.3.2.4-1: De EXOR-poort.

Als schakelingang B logisch 1 is, verschijnt het ingangssignaal geïnverteerd op de uitgang. Als schakelingang B logisch 0 is verschijnt het ingangssignaal onveranderd op de uitgang.



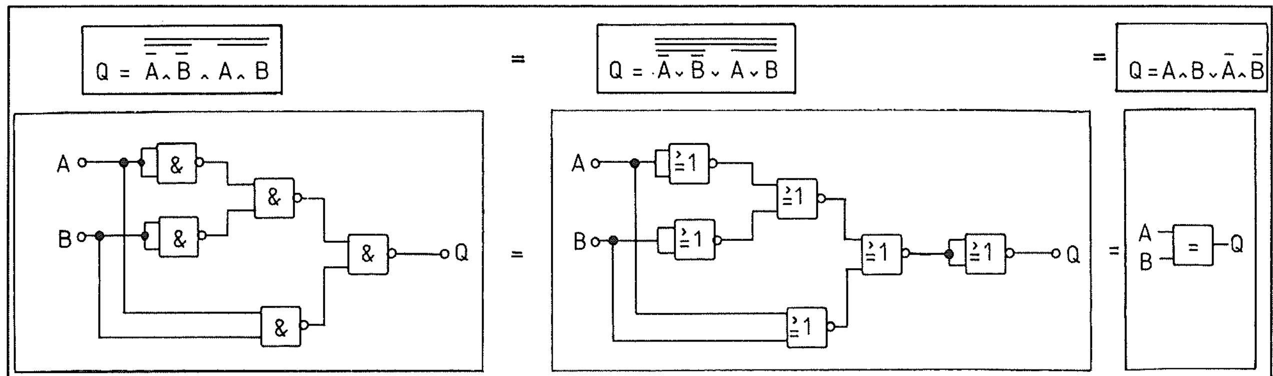
Figuur 3/6.3.2.4-2: Vier CMOS EX-OR-poorten, y = A + B = AB + AB.

3/6.3.2.5 EXCLUSIV-NOR-poorten.

Bij exclusieve NOR functie, kortweg EX-NOR of EXNOR genoemd, zal de uitgang slechts dan logisch 1 zijn, als beide ingangen aan elkaar gelijk zijn. Deze functie kan zowel met NAND-poorten worden verwezelijkt, als met NOR-poorten. In figuur 1 ziet U een schematische weergave van de eerste, terwijl figuur 2 de opbouw met NOR-poorten laat zien. De beide schakelingen van figuur 1 en 2 resulteren in dezelfde functietabel. De IC fabrikanten maken geen EX-NOR poorten. Om een wel verkrijgbare EX-OR functie om te toveren in een EX-NOR functie, is een inverter aan de uitgang noodzakelijk. De voorgestelde digitale functies werden slechts voor twee ingangen gemaakt. Het is mogelijk het aantal ingangen uit te breiden, zonder dat de functie van de poorten verandert.

Op de markt is een veelheid aan digitale IC's verkrijgbaar. Intern zijn zij altijd opgebouwd uit de basisfuncties. Het is een goede en voor het begrip zeer aan te bevelen oefening om op een oefenprint of breadboard zelf de functietabellen te controleren. In tabel 1 wordt voor diverse poorten de uitgangstoestand bij verschillende ingangscombinaties aangegeven. Als U van alle functies een IC koopt om mee te experimenteren, zult U waarschijnlijk nog geen tientje kwijt zijn, terwijl de IC's later altijd nog wel eens van pas komen.

6.3 Schakelingen van digitale functies



Figuur 3/6.3.2.5-1: EXNOR functie gerealiseerd met NAND-poorten.

Figuur 3/6.3.2.5-2: EXNOR functie gerealiseerd met NOR-poorten.

ingangen		uitgangstoestand															
A	B	Y1	Y2	Y3	Y4	Y5	Y6	Y7	Y8	Y9	Y10	Y11	Y12	Y13	Y14	Y15	Y16
0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
poort-functie:		Y=0	AND	Inhibit $a \wedge \bar{b}$	Y=A	Inhibit $\bar{a} \wedge b$	Y=B	EXOR	OR	NOR	EXNOR	Y=B	enable $a \vee \bar{b}$	Y=A	enable $\bar{a} \vee b$	NAND	Y=1

Tabel 3/6.3.2.5-1: Uitgangstoestand voor diverse functies bij bepaalde ingangscombinaties.

6.3 Schakelingen van digitale functies

3/6.4

De Booleaanse algebra

Inhoud

3/6.4.1 Inleiding
(verschenenin de 13e aanvulling)
3/6.4.2 Het herleiden van een functie vergelijking uit een schakeling
3/6.4.3 Het afleiden van de vergelijking uit functietabellen
3/6.4.4 Welke methode moet men nemen
3/6.4.5 Het tijddiagram
3/6.4.6 Hoe te vereenvoudigen
3/6.4.7 De grenzen van de Booleaanse algebra
3/6.4.8 De Booleaanse algebra voor de afgeleide poorten
3/6.4.9 De Booleaanse algebra voor geklokte logica

3/6.4.1

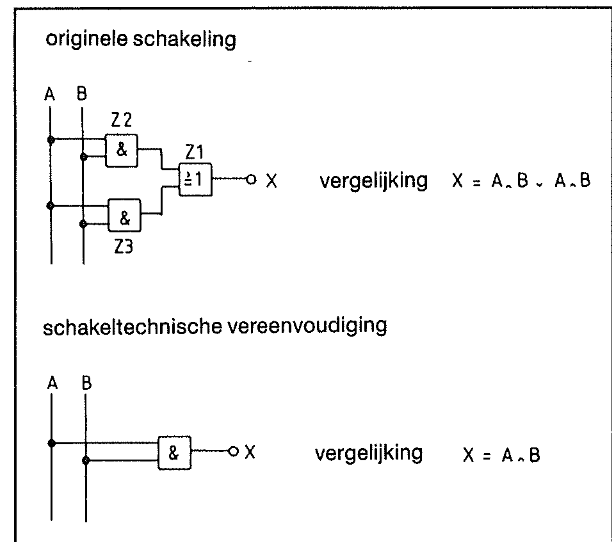
Inleiding

De Booleaanse algebra is een belangrijk hulpmiddel in de digitale techniek. Ten eerste om een schakeling in een zo algemeen mogelijke vorm algebraisch voor te stellen en in de tweede plaats om een schakeling te analyseren. Zelfs bij geklokte logica is gebruik van de Booleaanse algebra mogelijk. De allereerste studies van logica werden reeds gedaan door Aristoteles. Deze filosofeerde over zaken, die slechts 'waar' of 'niet waar' konden zijn, en dus slechts twee toestanden kenden. De wiskundige George Boole stelde in 1854 rekenregels op, gebaseerd op deze logica. Door het gebruik, dat wij ervan maken, wordt de Booleaanse algebra ook wel schakelalgebra genoemd.
Met behulp van de door Boole opgestelde rekenregels, is het mogelijk een ontworpen schakeling te optimaliseren, of uit de functietabel een functiediagram van de schakeling te destileren. In tabel 1 vindt u een overzicht van de rekenregels van de Booleaanse algebra.
Het basisprincipe van de Booleaanse algebra is het feit, dat bij het uitlezen van de functietabel of schakeling, de diverse benodige ingangssignalen in de vergelijking worden genoteerd als geïnverteerd of niet geïnverteerd. Laten we eerst eens proberen de functievergelijking uit een schakeling te destileren.

3/6.4.2

Het herleiden van een functie vergelijking uit een schakeling

Het volgende voorbeeld is eenvoudig gehouden, zodat het principe duidelijk naar voren komt. Het doel is de uitgangsvariabele X algebraisch uit te drukken. We volgen een logisch pad.



Figuur 3/6.4.2-1: Schakeling met twee ingangsvariabelen.

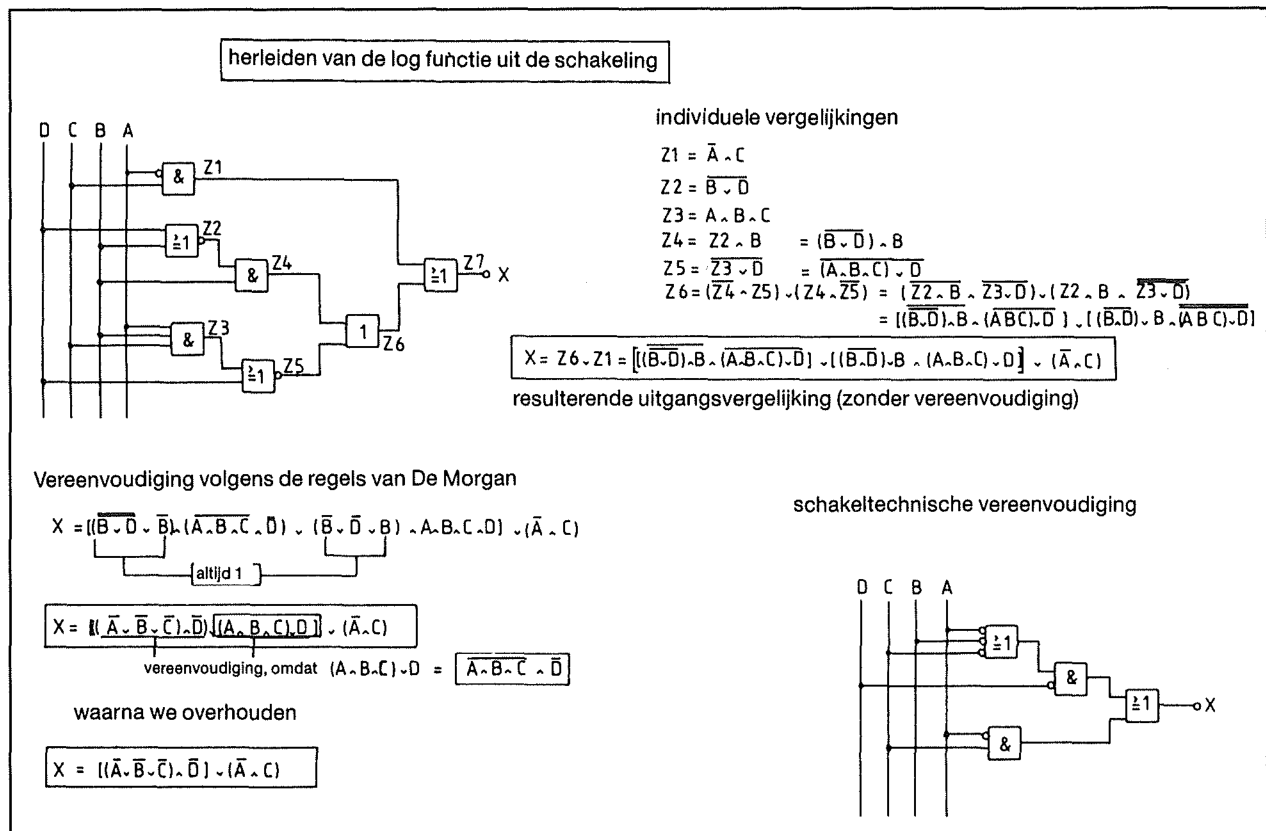
6.4 De Booleaanse algebra

Theorema's van de schakelalgebra

basisvergelijkingen		toelichting
$A \wedge 0 = 0$ $A \wedge 1 = A$ $A \wedge A = A$ $A \wedge \bar{A} = 0$	$A \vee 0 = A$ $A \vee 1 = 1$ $A \vee A = A$ $A \vee \bar{A} = 1$	$\bar{\bar{A}} = A$ –[1]o–[1]o– = [1] $\bar{\bar{\bar{A}}} = \bar{A}$ –[1]o–[1]o[1]o = [1]o– $\bar{A} = 0$ $A = 1$
uitbreidingen		
$A = A \wedge A = A \wedge A \wedge A$ $A = A \wedge 1 = A \wedge 1 \wedge 1$	$A = A \vee A = A \vee A \vee A$ $A = A \vee 0 = A \vee 0 \vee 0$	zelf verder uit te breiden
commutatieve eigenschap		
	$A \wedge B = B \wedge A$ $A \vee B = B \vee A$	verwisseling
associatieve eigenschap		
$(A \wedge B) \wedge C = A \wedge (B \wedge C) = B \wedge (A \wedge C) = A \wedge B \wedge C$ $(A \vee B) \vee C = A \vee (B \vee C) = B \vee (A \vee C) = A \vee B \vee C$		samenvoeging
absorptieregels		
$A \wedge (A \vee B) = A$ $A \wedge (\bar{A} \vee B) = A \wedge B$ $(A \vee B) \wedge (\bar{A} \vee B) = B$	$A \vee (A \wedge B) = A$ $A \vee (\bar{A} \wedge B) = A \vee B$ $A \vee (\bar{A} \wedge \bar{B}) = A \vee \bar{B}$ $\bar{A} \vee (A \wedge B) = \bar{A} \vee B$ $\bar{A} \vee (A \wedge \bar{B}) = \bar{A} \vee \bar{B}$ $(A \wedge B) \vee (\bar{A} \wedge B) = B$	vermenigvuldigen en combineren geeft de oplossing
distributieve eigenschap		
$(A \wedge B) \vee (A \wedge C) = A \wedge (B \vee C)$ $(A \vee B) \wedge (A \vee C) = A \vee (B \wedge C)$		verdeling
De Morgans wetten		
$A \wedge B \wedge C \wedge \ldots Z = A \wedge B \wedge C \wedge \ldots Z$ $A \vee B \vee C \vee \ldots Z = A \vee B \vee C \vee \ldots Z$		

Tabel 3/6.4.2-1: De rekenregels van de Booleaanse algebra.

6.4 De Booleaanse algebra



Figuur 3/6.4.2-2: Voorbeeld van vereenvoudiging van een schakeling met meerdere ingangsvariabelen.

Evenals bij een doolhof is het eenvoudiger de weg van de uitgang naar de ingang te vinden als omgekeerd. In het voorbeeld van figuur 1 hebben we te maken met een logische schakeling met twee ingangsvariabelen (A en B). Vanaf de uitgang X, komen we eerst OR-poort Z1 tegen. De ingangsvariabelen van Z1 zijn de uitgangen van AND-poorten Z2 en Z3, die op hun beurt de ingangsvariabelen A en B op hun ingangen zien.
Men kan dit als volgt noteren:

	X = Z1
waarbij	Z1 = Z2 V Z3
met	Z2 = A ∧ B Z3 = A ∧ B
zodat	X = (A ∧ B) V (A ∧ B)

volgens de distributieve eigenschap kan dit worden omgezet in:

X = A ∧ (B V B), waarin
B V B = B

zodat X = A ∧ B

Deze uitdrukking is de oplossing, daar zij met geen der regels van de Booleaanse algebra verder te vereenvoudigen is.

Als u het niet gelooft, bouwt u dan de schakeling eens na en u zult het zien. Bovendien kan dit het inzicht in de materie vergroten.

6.4 De Booleaanse algebra

Als men meer poorten aan elkaar knoopt neemt de hoeveelheid werk toe, maar het principe blijft gelijk. In figuur 2 zien we zo'n schakeling met meerdere poorten. Elke poort wordt eerst genoteerd met zijn ingangsvariabelen.
Vervolgens neemt men de functies van alle poorten op in een totaal-vergelijking. Vergeet niet, de haken om de OR-functies te zetten, daar anders AND-functies eerst worden uitgewerkt. Door toepassen van de regels kan de vergelijking worden vereenvoudigd. In figuur 2 is de uiteindelijke vereenvoudiging weer als een schakeling weergegeven. U ziet, dat de functie, die oorspronkelijk met een schakeling, die 7 poorten telde was verwezenlijkt ook kan worden bereikt met slechts vier poorten.

3/6.4.3

Het afleiden van de vergelijking uit functietabellen.

Deze methode wordt veel vaker gebruikt, als de bovenstaande om de doodeenvoudige rede, dat meestal niet de schakeling als gegeven dient, maar de te vervullen functie. Om te proberen de gegeven functie in een vergelijking weer te geven kennen we twee methodes:

1. De mintermen-methode
We zijn geïnteresseerd in de uitgang X en wel in het bijzonder, als daar een logische 1 verschijnt. In het voorbeeld van figuur 1: De mintermen-methoden, zijn zes mintermen te vinden. De omschrijving mintermen slaat op de logische AND combinatie van alle voorkomende ingangsvariabelen (conjuncties) van een schakeling. Elke minterm op zich heeft als schakeling gezien slechts bij één combinatie van de ingangsvariabelen de logische waarde 1. Alle mintermen onder elkaar worden middels de logische OR combinatie samengenomen. Als in een minterm onder de ingangsvariabelen een logische nul voorkomt, wordt deze aangeduid met zijn inverse (omgekeerde waarde). De eerste minterm ziet er als volgt uit:

A	B	C	X	
0	0	1	1	$X = A \wedge B \wedge C$

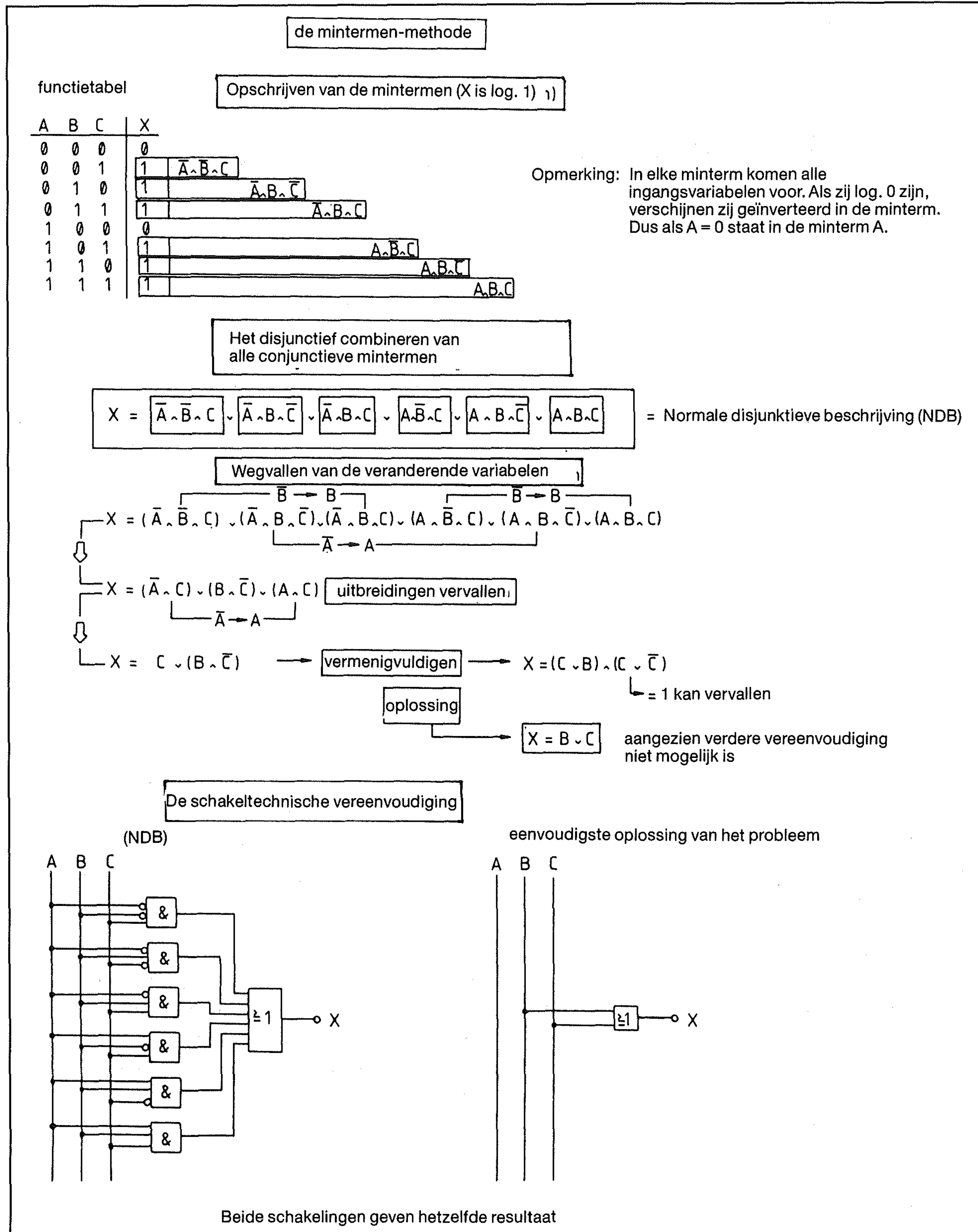
Dit is te beschrijven als X is A niet en B niet en C.

Opmerking: dikwijls stelt men een AND functie voor met een punt, die dan evenals in de gewone algebra weer wordt weggelaten. Voor de OR functie wordt dan het 'x' teken gebruikt. De nieuwste normen schrijven schrijven echter het gebruik van de tekens ∧ en v voor, voor resp. AND en OR.

2. De maxtermen-methode
Evenals bij de mintermen-methode is ook hier de functietabel het uitgangsgegeven voor de analyse van de schakeling. Echter nu zijn we geïnteresseerd in de logische 0 toestanden van de uitgang. Voor elke 0-toestand van de uitgang definiëren we een maxterm. Alle maxtermen worden conjunctief met elkaar verbonden. Het is hierbij van belang, de afzonderlijke maxtermen binnen haken te plaatsen. De AND functie heeft immers binnen de digitale rekenvolgorde voorrang op de OR-functie.

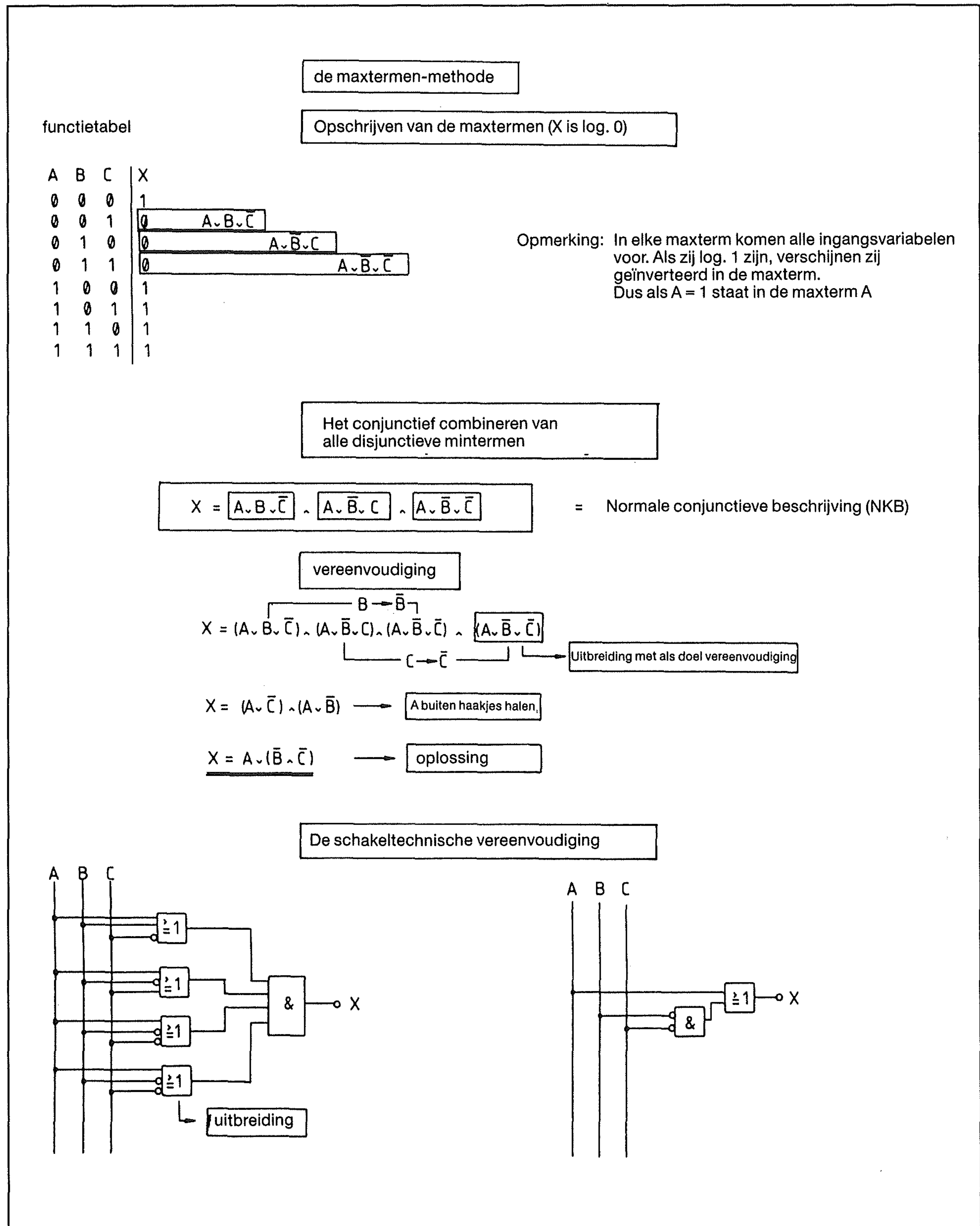
De maxterm is de OR-functie van alle voorkomende ingangsvariabelen, (disjunctie) van een schakeling. Iedere maxterm heeft als schakeling gezien, slechts bij één ingangscombinatie in de functie tabel de waarde 0.

6.4 De Booleaanse algebra



Figuur 6.4.3-1: De mintermen-methode

6.4 De Booleaanse algebra



Figuur 3/6.4.3-2: De maxtermen-methoden.

6.4 De Booleaanse algebra

Alle maxtermen worden onderling gecombineerd met AND logica.
Als binnen een maxterm een ingangsvariabele als logische 1 voorkomt, wordt deze variabele geïnverteerd. U vindt dit alles verduidelijkt in figuur 2.

3/6.4.4

Welke methode moet men nemen?

Daarop kan helaas geen eenduidig antwoord worden gegeven. Het beste kiest u een der beide methodes. Het juist tegengesteld inverteren bij de beide methodes werkt verwarrend als men ze door elkaar gebruikt. Door steeds dezelfde methode te gebruiken krijgt men ervaring en zal het steeds gemakkelijker gaan. Totdat u een soort automatisme hebt ontwikkeld, raad ik u aan het bij een methode te houden.

Voor de ervaren electronicus zal de keuze dikwijls afhangen van de functietabel. De toestand die het minst voorkomt wordt dan bepalend voor de te gebruiken methode. Men voorkomt dan, dat er zeer lange disjunctieve of conjunktieve vergelijkingen moeten worden opgelost, hetgeen doorgaans nogal tijdrovend is en het nodige inzicht vereist. Als men deze keuzeregel op het voorbeeld van figuur 1 in hoofdstuk 3/6.4.3. had toegepast, dan zou men zeker met de maxtermen-methode sneller tot een oplossing zijn gekomen.

6.4 De Booleaanse algebra

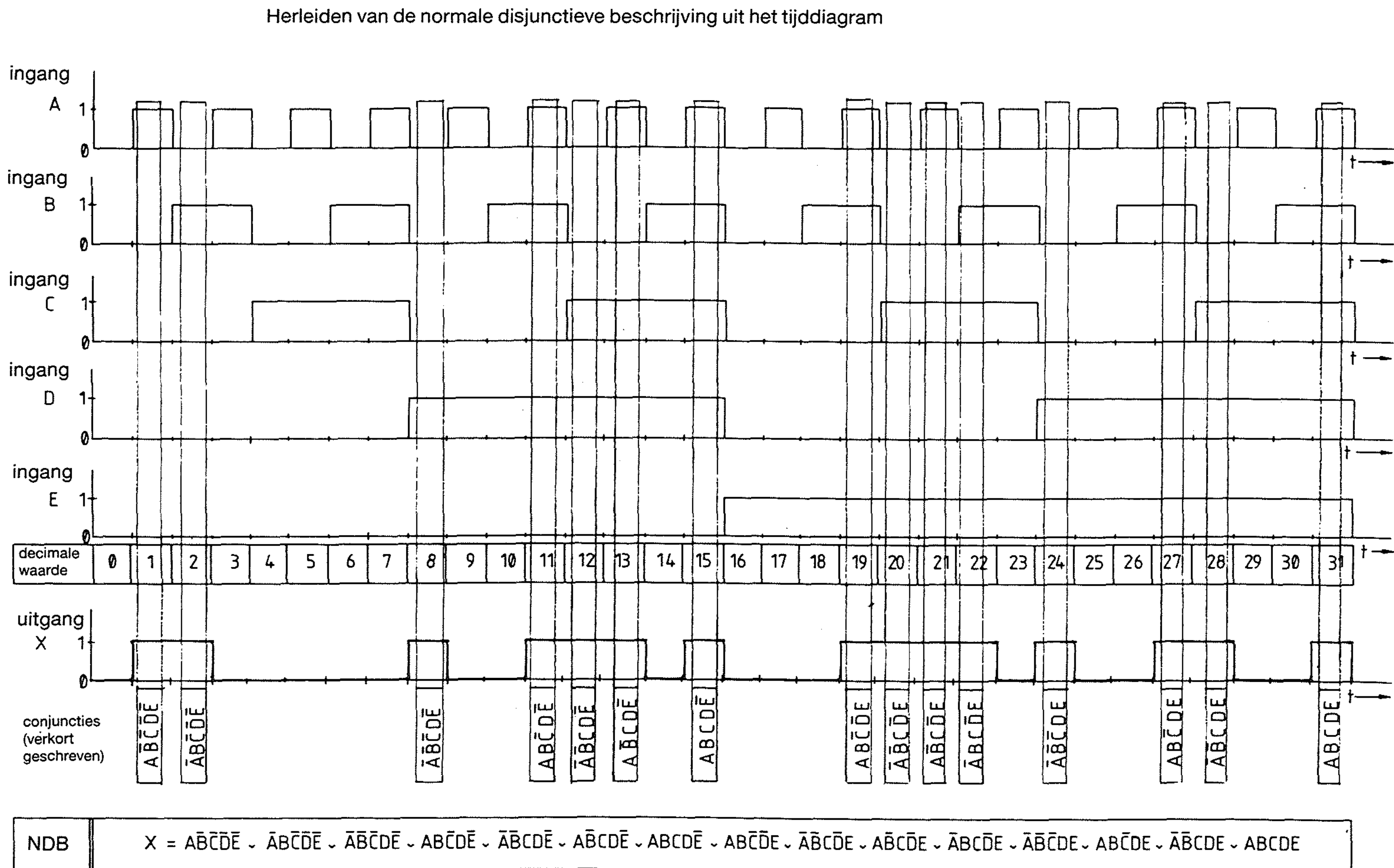
3/6.4.5

Het tijddiagram als gegeven om de functietabel samen te stellen

Een tijddiagram is een tweedimensionale weergave van een functietabel. Op de Y-as staat de spanning, of het logische niveau, op de X-as is de tijd uitgezet. Op de tijdsas vinden we de onderlinge betrekkingen terug. Men kan dit op een oscilloscoop zichtbaar maken. Het probleem is meestal, dat men op een oscilloscoop vaak maar een of twee kanalen ter beschikking heeft, waardoor niet alle in- en uitgangen tegelijkertijd kunnen worden bestudeerd. Logic-analysers kunnen dit wel, maar dit zijn apparaten, waarvan men de mogelijkheden dan ook duidelijk aan de prijs kan aflezen.

Voor de amateur, die slechts over een enkel, of een dubbelkanaals oscilloscoop beschikt, blijft over om goed te triggeren en de signalen een voor een over te tekenen op millimeterpapier. In digitale schakelingen triggert men het best extern op de traagste frequentie. Natuurlijk mag de tijdbasis niet worden veranderd gedurende de metingen en moet het overtekenen op millimeterpapier nauwkeurig gebeuren.
Het destileren van de logische functie kan weer op twee manieren. In figuur 1 bekijken we een uitgang met de erbij behorende ingangen. Triggering is gedaan met het signaal van ingang E. Voor het bepalen van de logische functie werd de mintermen-methode gekozen. (De plaatsen in het tijddiagram, waar de uitgang de logische waarde 1 heeft). Het aantal plaatsen waar deze waarde voorkomt bepaald dus het aantal conjuncties, dat disjunctief gecombineerd wordt. Het volgende hoofdstuk geeft een beschrijving van de vereenvoudiging.

6.4 De Booleaanse algebra



Het vereenvoudigen van deze vergelijking is een geduld werkje. Als men de resulterende schakeling in werkelijkheid wil bouwen, is de methode die de minste componenten vraagt een programmeerbare code-omzetter, bijv. een EPROM.

Figuur 6.4.5-1: Het herleiden van de functietabel uit het tijddiagram.

3/6.4.6

Hoe te vereenvoudigen

Het vereenvoudigen kan met beide methodes gebeuren. Het enige hulpmiddel dat u nodig hebt is een geoefend oog. Na het opstellen van de mintermen of maxtermen begint men elke term te vergelijken met zijn buren op het gelijk zijn van minstens n-1 ingangsvariabelen (waarbij n het totale aantal ingangs-variabelen voorstelt) en op het tegengesteld zijn van slechts een variabele binnen termen die aan de eerste voorwaarde voldoen. Beide termen moeten hetzelfde aantal ingangsvariabelen hebben. Als dat het geval is, dan worden beide termen samengenomen, waarbij de tegengestelde variabelen wegvallen, waardoor deze termen in de volgende vergelijking 1 variabele minder hebben. Als het aantal termen dat valt te combineren oneven is, dan kan een reeds gebruikte term nogmaals als uitbreiding worden toegevoegd. De term moet natuurlijk wel echt voorkomen, men kan dus niet zomaar een willekeurige term toevoegen. Men kan met deze methode doorgaan, tot alle mogelijkheden zijn uitgeput. Dus eerst vergelijken met de directe buren, dan met de daaropvolgende (nog niet gebruikte) termen enz.
De overige termen (die niet vallen te combineren) gaan we met een algebraïsch oog bekijken, om te zien of er geen delen ervan buiten haken gebracht kunnen worden. Hierna proberen we weer de resterende termen samen te voegen. Pas op met de optredende verwisselingen van con- en disjunctie.

De inhoud van de haken wordt weer onderzocht op de mogelijkheid van uitbreiden en samenvoegen.

Als er geen mogelijkheden meer zijn vereenvoudigingen of samenvoegingen uit te voeren, dan heeft men de eenvoudigste schakeling gevonden, die de gewenste uitgangsfunctie realiseert. Bestudeer voor verduidelijking figuur 1. Bij de praktische opbouw van de schakeling hoeft men voor vaker in de schakeling voorkomende logische (deel-)functies niet afzonderlijke poorten te nemen. De meeste standaard TTL-poorten kunnen namelijk zonder bezwaar met meerdere TTL-ingangen worden belast. Bij gewone TTL wel tot 10 stuks. Als men de gevonden schakeling nabouwt, kan men controleren, of het inderdaad werkt zoals veracht, of dat er ergens fouten in geslopen zijn.

6.4 De Booleaanse algebra

Voordeel:
Er is gebruik gemaakt van poorten met niet meer, dan 3 ingangen. Hierdoor is het mogelijk de schakeling met standaard in de handel verkrijgbare IC's te maken.

Figuur 6.4.6-1: Vereenvoudiging van de NDB en de schakeltechnische oplossing.

3/6.4.7

De grenzen van de Booleaanse algebra

Als systeem zijn er geen grenzen aan de Booleaanse algebra. Als het aantal ingangsvariabelen stijgt, kan echter het overzicht verloren gaan, omdat de termen veel te lang worden, zoals men al bij het voorbeeld uit het vorige hoofdstuk kan zien. Vermeldenswaart is, dat hoe dan ook de normale disjunctieve of konjunctieve beschrijving blijft functioneren. De echte grenzen liggen in het geduld van de gebruiker.

De Booleaanse algebra is een goede en snelle methode om voor een beperkt aantal ingangsvariabelen een algebraïsche vergelijking te vinden, of deze op te stellen voor logische poorten. De Booleaanse algebra levert altijd op de snelste manier de eenvoudigste schakeling voor een gegeven probleem op.

6.4 De Booleaanse algebra

3/6.4.8

De Booleaanse algebra voor de afgeleide poorten

De afgeleide poorten, dit zijn de poorten die een functie hebben, die alleen door combinatie van basisfuncties zijn te verwezenlijken, hebben bijna altijd aan de in- of uitgangen inverters. Deze inverters zorgen ervoor, dat een ingangs-signaal, of het uitgangssignaal in de tegengestelde waarde wordt omgezet. Natuurlijk moet de Booleaanse algebra ook hier toepasbaar zijn.

Een voorbeeld: Geef de functie van een NAND-poort. Zoals inmiddels bekend, is een NAND-poort samengesteld uit de basisfuncties 'AND' en 'INVERT'. De functie van de AND-poort is:

$X = A \wedge B$

Dit wordt gevolgd door de invertering, die wordt aangeduid, door over de hele uitdrukking een streep te zetten, dus:

$X = \overline{A \wedge B}$, (X is A and B not).

Elke invertering wordt aangegeven door een streep boven de variabele, of de hele uitdrukking. (Bij positieve logika). Als een schakeling, waarin afgeleide poorten voorkomen moet worden geanalyseerd, dan moeten deze poorten eerst naar poorten met basisfuncties worden getransformeerd. Hoe dit wordt gedaan komt later nog aan bod.

6.4 De Booleaanse algebra

3/6.4.9

De Booleaanse algebra voor geklokte logika

Ik kan me voorstellen, dat uw eerste vraag is: 'Wat is geklokte logika?'. Hoewel dit later nog aan de orde komt, lichten we even een tipje van de sluier op. Geklokte logika, dus geklokte schakelingen, zijn schakelingen, waarvan de uitgang pas dan op de ingangssignalen reageert, nadat op een andere ingang een puls is verschenen. (bijv. Set en Reset van een D-flip-flop.) De ingang, waarop de puls verschijnen moet wordt doorgaans de klokingang, of kortweg de klok genoemd.

Booleaanse algebra in geklokte schakelingen

Soort flip-flop	karakteristieke vergelijking
RS	$Q_{n+1} = [S \vee (R \wedge Q_n)]$
JK	$Q_{n+1} = (J \wedge \overline{Q}_n) \vee (\overline{K} \wedge Q_n)$
D	$Q_{n+1} = D_n$
T	$Q_{n+1} = (T \wedge \overline{Q}_n) \vee (\overline{T} \wedge Q_n)$

Q_{n+1} uitgangstoestand na de klokpuls

Q_n : uitgangstoestand voor de klokpuls

Tabel 3/6.4.9-1: Booleaanse algebra voor geklokte schakelingen.

De behandeling van deze sequenciële schakelingen komt pas later aan de orde. Hier wordt alleen even ingegaan op de wijze, waarop deze functies in de Booleaanse algebra worden uitgedrukt.
Tabel 1 geeft een overzicht van de Booleaanse algebra voor geklokte logika aan de hand van de meest voorkomende flipflop types. De aanduiding t+1 slaat op het tijdstip direct na de klokpuls. Pas op dat moment komt de uitgangstoestand overeen met de erbij behorende ingangstoestand. Voor de klokpuls komt de uitgangstoestand overeen met de ingangscondities die golden voor de vorige klokpuls.
De uitgang is algemeen weergegeven als functie van de karakteristieke functie van de flip-flop. De functie is echter pas waar, na de klokpuls.
De getoonde functies zijn niet te vereenvoudigen, daar zij de basisfunctie van de schakeling voorstellen. Gezien de invloed van de klok, zal het duidelijk zijn, dat deze cruciaal is bij de analyse van deze schakelingen. Geklokte schakelingen komt men tegen in bijv. schuifregisters, tellers ed. Voor de vereenvoudiging of uitdrukking van de functies van dergelijke schakelingen gebruikt men zogenaamde KV-tafels, die in een volgend hoofdstuk worden besproken.

6.4 De Booleaanse algebra

3/6.5

De KV-diagrammen

Bij de ontwikkeling van digitale schakelingen is de grondwet altijd de schakeling zo gecompliceerd te maken als nodig is en zo eenvoudig als mogelijk is. Het is dus altijd de bedoeling het gewenste resultaat te bereiken met een minimum aan inspanning.

Om dit doel te bereiken is de Booleaanse algebra ontwikkeld. Het gewenste resultaat, nl. een minimum aan inspanning blijft bij gebruik van de Booleaanse rekenregels echter nog wel eens uit. De regels zijn tamelijk ingewikkeld en vergen veel schrijfwerk. Schrijfwerk, dat van dien aard is, dat een foutje makkelijk wordt gemaakt.
Het ligt dus voor de hand dat men op zoek is gegaan naar andere mogelijkheden tot minimalisering van digitale schakelingen. De heren Veitch en Karnaugh hebben zich hiermee beziggehouden. Zij ontwikkelden een methode, die tot op heden de snelste weg levert voor vereenvoudiging van digitale functies. De methode is gebaseerd op diagrammen of tabellen, die naar de ontwikkelaars KV-diagrammen worden genoemd. Met behulp van KV-diagrammen is het bijna altijd mogelijk de kleinste schakeling te vinden. Uitzonderingen bevestigen echter ook hier de regel.

De samenstelling van een KV-diagram

De samenstelling van een KV-diagram is een ruimtelijke voorstelling van de functie. De samenstelling is afhankelijk van de functietabel, nauwkeuriger gezegd van het aantal mogelijke ingangscombinaties. Een functietabel met 5 ingangsvariabelen zal 32 ingangscombinaties hebben en is dus duidelijk gecompliceerder, dan een functie met maar twee ingangsvariabelen.
Men begint met voor elke denkbare (mogelijke) ingangscombinatie een veld te reserveren, waarin verderop in het proces de uitgangsgrootheid X wordt genoteerd, onafhankelijk van de logische toestand daarvan. Het KV-diagram is een twee-dimensionale weergave van de functietabel in een vorm, die op een schaakbord lijkt.

De veldaanduiding van een KV-diagram

Om aan elk veld in een KV-diagram een gedefiniëerde waarde uit de functietabel te kunnen toekennen, moet elk veld met ingangscombinaties of veldaanduidingen eenduidig kunnen worden aangesproken. Er mogen dus geen verboden combinaties (deze zijn niet gedefiniëerd), of dubbele aanduiden voorkomen.

6.5 KV-diagrammen

De veldaanduiding zou er uit kunnen zien als een der voorbeelden in figuur 3/6.5-1. Natuurlijk zijn er nog wel andere mogelijkheden om een kader om het diagram te plaatsen. Wat is de meest zinvolle mogelijkheid?

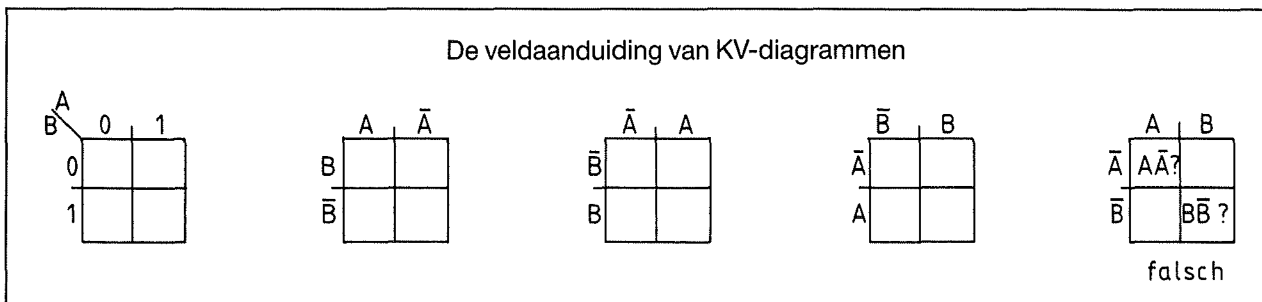
De praktijk heeft uitgewezen, dat de eerste mogelijkheid weliswaar snel is te noteren, echter alleen bruikbaar is bij positieve logika (waarvoor hoog + is). Over het algemeen is het tweede voorbeeld beter. Als het diagram groter wordt dan zestien velden, blijft hierbij toch het overzicht bewaard, ongeacht de gebruikte logika. De vijfde variant is foutief. Twee velden krijgen een verboden combinatie toegewezen terwijl twee andere gedefiniëerde ingangscombinaties niet voorkomen. Dubbele aanduiding op de randen moet men achterwege laten, daar dit bij grotere KV-diagrammen zeker tot verwarring zal leiden.

KV-diagrammen kunnen het best vierkant worden gemaakt. Alleen bij combinaties van drie, zes enz. ingangsvariabelen is dit niet mogelijk. Hoe de afzonderlijke KV-diagrammen voor verschillende aantallen ingangsvariabelen worden gemaakt is te zien in figuur 4/6.5-2. Er moet worden gezorgd, dat elke mogelijke ingangscombinatie een veld heeft.

Het invullen van KV-diagrammen

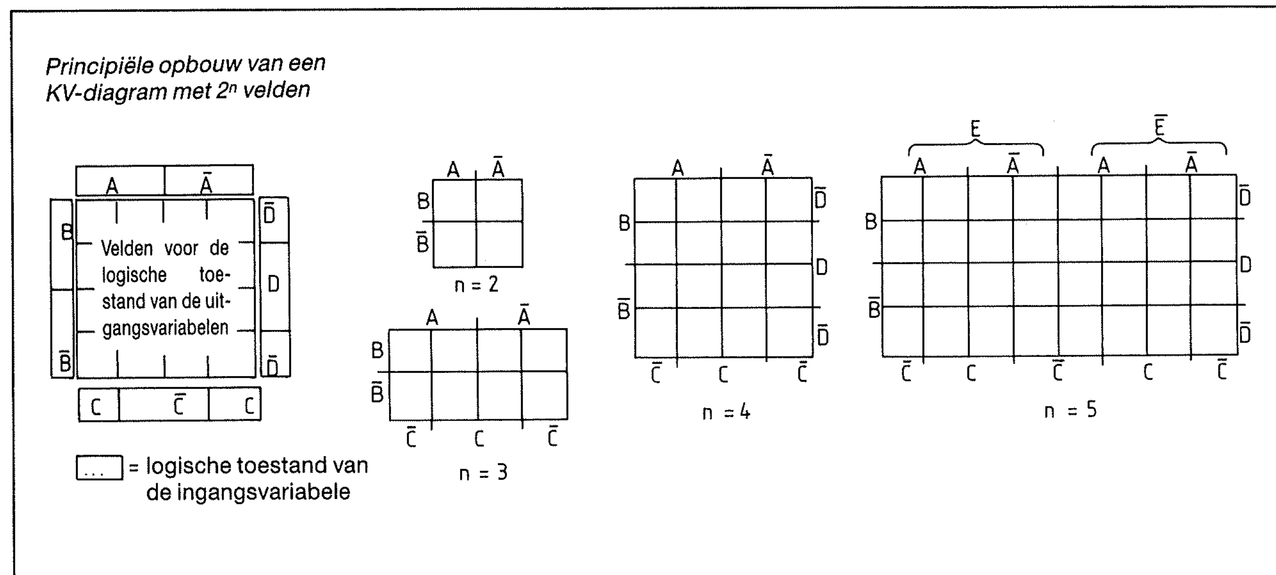
Nadat het KV-diagram van eenduidige veldaanduidingen is voorzien kunnen we aanvangen met het invullen. Uitgaande van de functietabel is dit in de meeste gevallen een routineklusje. De uitgangstoestanden bij alle ingangscombinaties zijn van belang. Zij worden in de betreffende velden van het KV-diagram ingevuld. In principe is de volgorde onbelangrijk. Het is echter een goede gewoonte de functietabel van boven naar onder te doorlopen, omdat daarmee wordt voorkomen, dat er combinaties worden overgeslagen. Als alle combinaties doorlopen zijn, moeten alle velden van het KV-diagram ingevuld zijn.

Bekijken we in figuur 3/6.5-2 het plaatje met n=3. Daar moeten dus acht velden worden ingevuld. In het veld linksboven komt de uitgangstoestand, die hoort bij A=1, B=1 en C=0 (dan is immers $\bar{C}$=I). Op de velden van de bovenste rij vullen we van links naar rechts de uitgangstoestanden in, die horen bij de volgende ingangscombinaties:
A=1, B=1, C=0 / A=1, B=1, C=1 / A=0, B=1, C=1 / A=0, B=1, C=0
Op de velden van de onderste rij vullen



Figuur 3/6.5-1: Mogelijke veldaanduiding van KV-diagrammen.

6.5 KV-diagrammen



Figuur 3/6.5-2: Opbouw van KV-diagrammen met n-ingangsvariabelen.

we evenzo de uitgangstoestanden in, die horen bij de volgende ingangscombinaties:
A=0, B=0, C=0 / A=1, B=0, C=1 / A=0, B=0, C=1 / A=0, B=0, C=1 / A=0, B=0, C=0

Als men met het invullen vertrouwd is geraakt, kan men het invulprincipe veranderen. De ervaren ontwerper zoekt de minder vaak optredende uitgangscombinaties uit, schrijft deze doelbewust in het KV-diagram en vult de rest in met de overige logische uitgangsvariabelen, of laat deze zelfs geheel weg.

Een bijzonder geval doet zich voor bij de zogenaamde redundante combinaties. Redundante combinaties komen voor, wanneer voor het samenstellen van de uitgangsvariabelen niet alle mogelijke ingangscombinaties nodig zijn. We verduidelijken dit met een voorbeeld:

Een binaire teller bestaat uit vier flipflops en kan dus tellen tot $2^4=16$. Echter de teller moet bij stand $10_{(10)}$ het tellen afbreken en kan opnieuw beginnen. De waarde $10_{(10)}$ mag echter niet verschijnen. De zestien mogelijke combinaties zijn 0 t/m $15_{(10)}$. Omdat we bij $10_{(10)}$ ophouden komen slechts de volgende binaire uitgangscombinaties voor: $000_{(2)}$ tot $1001_{(2)}$. De combinaties $1010_{(2)}$ tot $1111_{(2)}$ blijven dus ongebruikt. Deze zijn redundant, dus teveel. Voor deze redundante combinaties vindt men in de functietabellen vaak een "x" of "d" aanduiding. Deze redundante combinaties worden echter in de KV-diagrammen wel ingevuld. Bij de uiteindelijke bewerking van de KV-diagrammen, het samennemen kunnen ze naar believen worden geïnterpreteerd als logisch 1 of logisch 0, daar de schakeling niet van deze redundante combinaties afhanklijk is. De redundante uitgangscombinaties zullen in de echte schakeling immers nooit voorkomen. Zij kunnen echter bij de vereenvoudiging van de schakeling een belangrijke rol spelen.

6.5 KV-diagrammen

Bij de tot nu toe besproken benaderingen is steeds de functietabel uitgangsgegeven geweest. Natuurlijk kan het invullen van de KV-diagrammen ook rechtstreeks vanuit het tijddiagram gebeuren. Het wordt moeilijk, als alleen de schakeling ter beschikking staat. Als de uitgangsvergelijking in een disjunctieve of een conjunctieve vorm is weergegeven, dan kan met wat praktijkervaring het KV-diagram nog wel worden ingevuld. Afhankelijk van de ingewikkeldheid van de vergelijking, zal het meer of minder werk zijn. Deze wijze van werken wordt door ontwerpers van digitale schakelingen uiterst zelden toegepast. We zullen de bespreking dan ook achterwege laten.

Het samennemen en opstellen van de uitgangsvergelijking

Uitgaande van het ingevulde KV-diagram begint het eigenlijke werk. Door de manier waarop het KV-diagram is samengesteld verandert zowel in horizontale, als in vertikale richting van veld naar veld meestal slechts één ingangsvariabele. Dat geld ook over de rand heen. Men beschouwe daartoe het KV-diagram als op een bol geplakt, dat wil zeggen, dat men de velden op de linkerrand beschouwd als buren van de velden op de rechterrand en idem dito voor de velden van boven- en benedenrand.

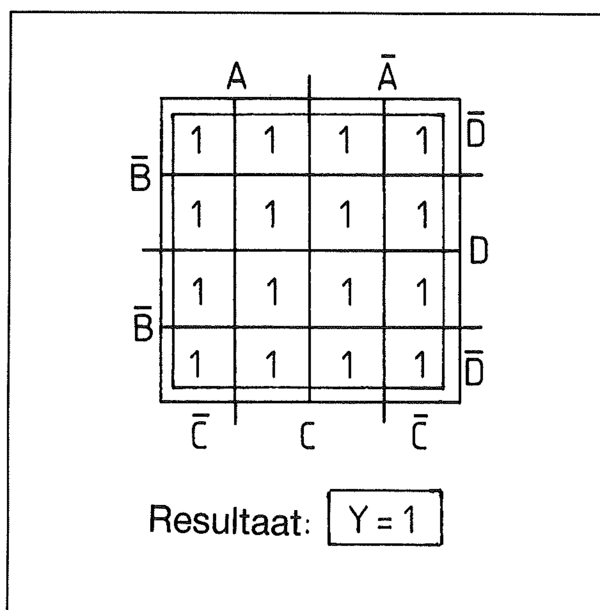
Als twee buurvelden dezelfde inhoud hebben, kunnen ze samengenomen worden. Bij het samennemen valt de veranderde ingangsvariabele weg, aangezien zijn verandering op de uitgangsvariabele geen invloed blijkt te hebben. De velden kunnen alleen in horizontale of vertikale richting worden gecombineerd. Diagonale velden kunnen niet worden samengenomen. Tot zover was de vereenvoudiging en het samennemen ook met de gewone Booleaanse algebra mogelijk geweest. Het KV-diagram levert echter het voordeel op, dat 2^n velden ineens kunnen worden samengenomen. Het op deze wijze samennemen, noemt men blokvorming. Het geheim van vereenvoudiging met KV-diagrammen ligt juist in deze blokvorming. Het samennemen is hier aanzienlijk eenvoudiger en dus sneller, dan met behulp van Booleaanse algebra.

Blokvoming in KV-diagrammen

Bij de blokvorming kan zowel de mintermen als de maxtermen methode worden gebruikt. Men kan dus zowel de logisch 1 velden, als de logisch 0 velden bewerken. De keuze is afhankelijk van de minst voorkomende toestand. Aangezien het KV-diagram met 16 velden in de praktijk het meest voorkomt is dit als uitgang genomen voor de verdere uitleg van de benadering van KV-diagrammen.

Het ingevulde KV-diagram wordt onderzocht op de mogelijkheid om blokken te vormen. Men zoekt eerst de grootste blokken. Het grootst mogelijke blok zou alle zestien velden omvatten, als de logische toestand daarvan voor alle velden gelijk is (zie figuur 3/6.5-3). De logische toestand van alle velden is dan immers hetzij 1, hetzij 0. Een dergelijke toestand komt uiterst zelden voor. Mocht zij zich voordoen, dan maakt een ervaren ontwerper niet eens een KV-diagram, maar haalt de vergelijking direct uit de functietabel.

6.5 KV-diagrammen



Figuur 3/6.5-3: Een 16-velden blok

Een groot aantal redundante uitgangscombinaties levert vaak een 16-velden blok op, b.v. bij een synchrone voorwaartsteller, die is samengesteld met JK-flip-flops verschijnen in de acht lagere tellerstanden acht redundante uitgangscombinaties en acht logische 1 locaties, wat leidt tot gebruik van een 16-velden blok (figuur 3/6.5-4).

Als men vaststelt, dat het niet mogelijk is een 16-velden blok te maken, dan moet men de mogelijkheid van een 8-velden blok onderzoeken. 8-Velden blokken kunnen in een 16-velden KV-diagram op verschillende plaatsen voorkomen (figuur 3/6.5-5). De verwerking van 8-velden blokken is al niet zo eenvoudig meer. Bij de vereenvoudiging vallen slechts drie ingangsvariabelen weg en er blijft er telkens een over. Deze overblijvende ingangsvariabele moet in de eindvergelijking worden meegenomen.

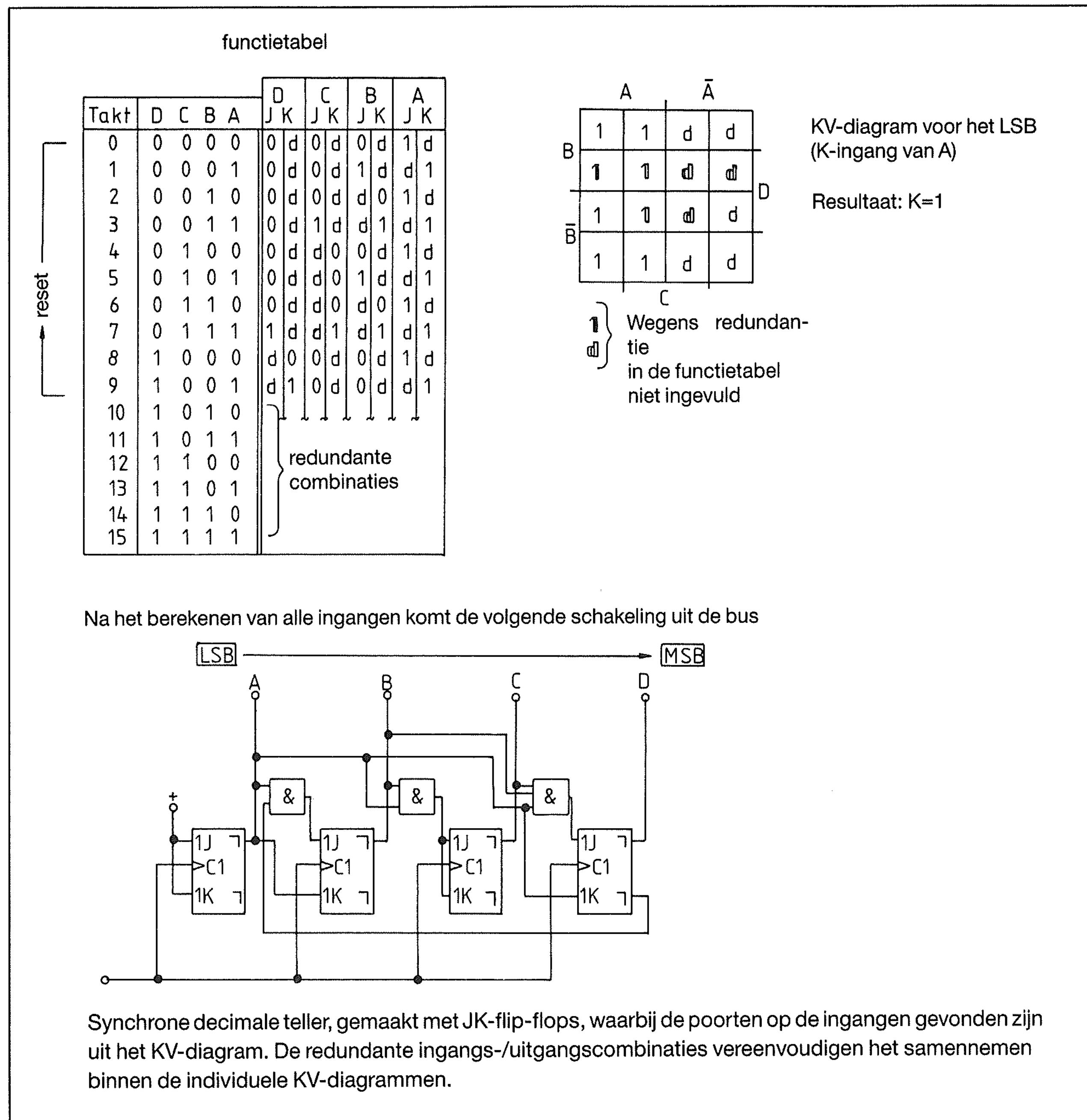
Als zelfs 8-velden blokken niet te vinden zijn, dan kan men niet gaan zoeken naar blokken van 14, 12, 6 of 3 velden, daar die niet aan de voorwaarde voldoen machten van 2 te zijn (2^n, waarbij n-1, 2, 3, ...). De volgende mogelijkheid is dus de 4-velden blokken.

Als voorbeeld voor het grote aantal 4-velden blokken, dat mogelijk is, hebben we er in figuur 3/6.5-6 een aantal aangeduid, die wat minder voor de hand liggen, omdat ze over de rand liggen. Door het maken van 4-velden vallen twee ingangsvariabelen weg. De laatste mogelijkheid voor het maken van combinaties zijn de 2-velden blokken. Bij een 2-velden blok valt er slechts een variabele weg. Dat is ook het geval bij de Booleaanse algebra. 2-velden blokken worden dikwijls overlappend genomen (zie figuur 3/6.5-7).

Bij het samennemen, blijven dikwijls enkele velden over, die niet samen te nemen zijn, omdat ze geen buurman hebben met dezelfde logische toestand. Deze velden moeten als conjunctie of als disjunctie onverkort uit het KV-diagram worden overgenomen.

Als men het KV-diagram volgens de mintermen methode uitleest, dan moet elk veld, waarin een logische 1 voorkomt aan de beurt zijn geweest. (Bij de max-termen methode moettten dat de velden waarin een logische 0 voorkomt zijn). Voor het samenstellen van blokken mag een veld meermalen worden gebruikt om steeds het grootste mogelijke blok te nemen. Het is beter een 4-velden blok te maken, als 2 twee velden blokken. Uit de praktijk blijkt, dat de beste methode is, meerdere malen te

6.5 KV-diagrammen



functietabel

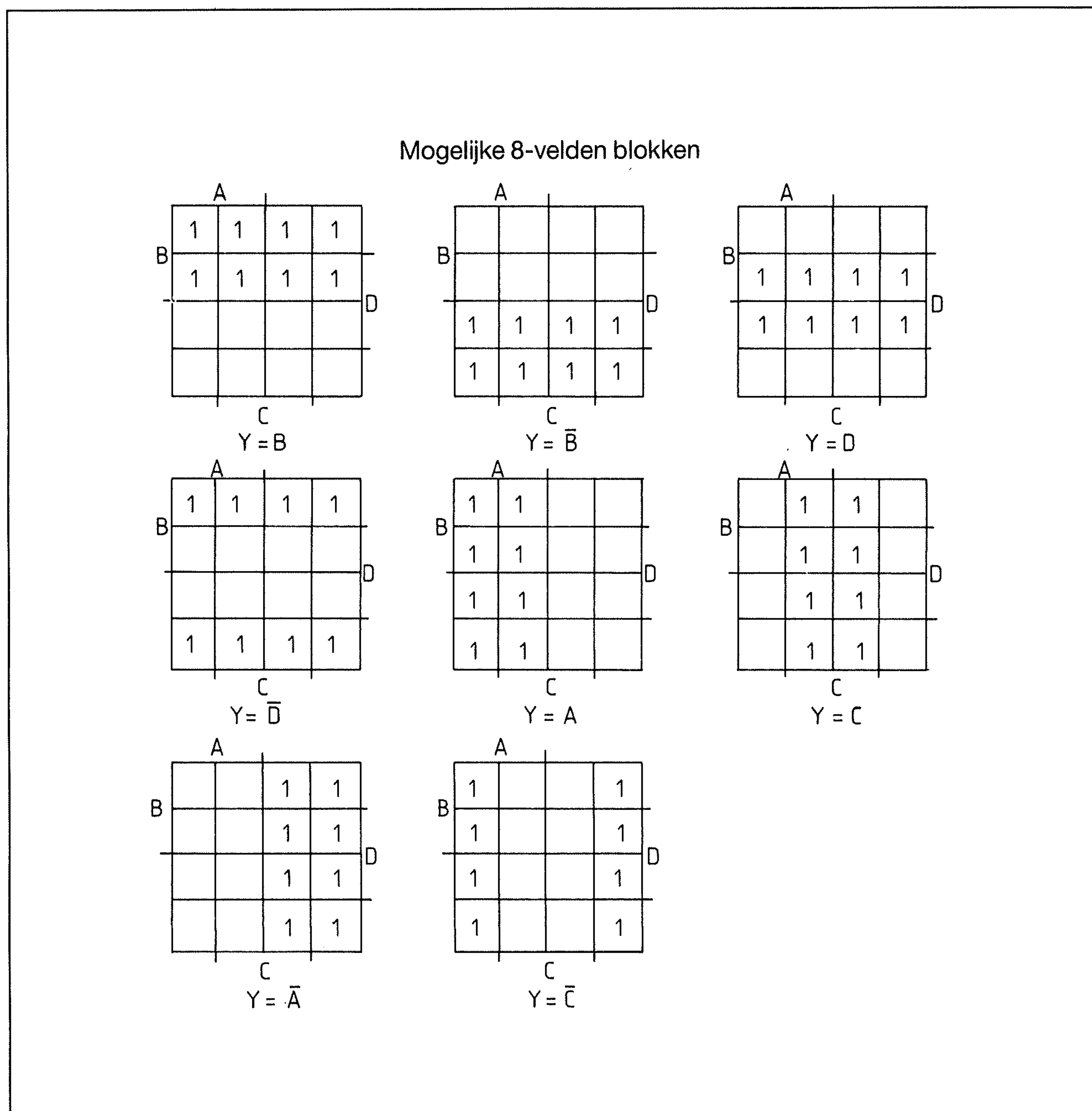
Takt	D C B A	D J	D K	C J	C K	B J	B K	A J	A K
0	0 0 0 0	0	d	0	d	0	d	1	d
1	0 0 0 1	0	d	0	d	1	d	d	1
2	0 0 1 0	0	d	0	d	d	0	1	d
3	0 0 1 1	0	d	1	d	d	1	d	1
4	0 1 0 0	0	d	d	0	0	d	1	d
5	0 1 0 1	0	d	d	0	1	d	d	1
6	0 1 1 0	0	d	d	0	d	0	1	d
7	0 1 1 1	1	d	d	1	d	1	d	1
8	1 0 0 0	d	0	0	d	0	d	1	d
9	1 0 0 1	d	1	0	d	0	d	d	1
10	1 0 1 0								
11	1 0 1 1								
12	1 1 0 0								
13	1 1 0 1								
14	1 1 1 0								
15	1 1 1 1								

Synchrone decimale teller, gemaakt met JK-flip-flops, waarbij de poorten op de ingangen gevonden zijn uit het KV-diagram. De redundante ingangs-/uitgangscombinaties vereenvoudigen het samennemen binnen de individuele KV-diagrammen.

Figuur 3/6.5-4: Synchrone voorwaartsteller met JK-flip-flops

starten. De eerste keer moet men proberen zoveel mogelijk velden te omcirkelen, zonder andere velden dubbel te gebruiken. Als er losse velden overblijven, moet men proberen, die zogoed mogelijk te gebruiken. De redundante combinaties kunnen daarbij helpen. Als uiteindelijk redundante combinatie overblijven, dan moeten die niet worden uitgelezen. Zij blijven gewoon staan.

6.5 KV-diagrammen

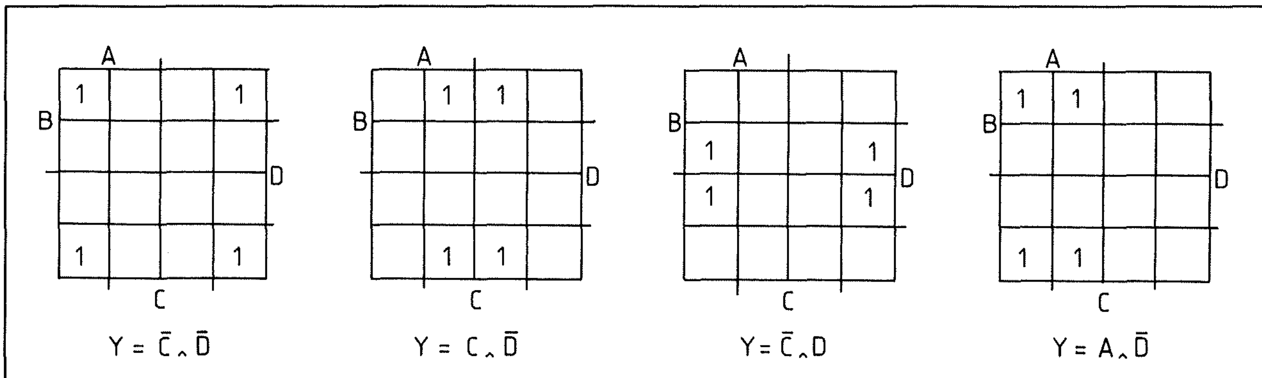


Figuur 3/6.5-5: Mogelijke 8-velden blokken in het KV-diagram

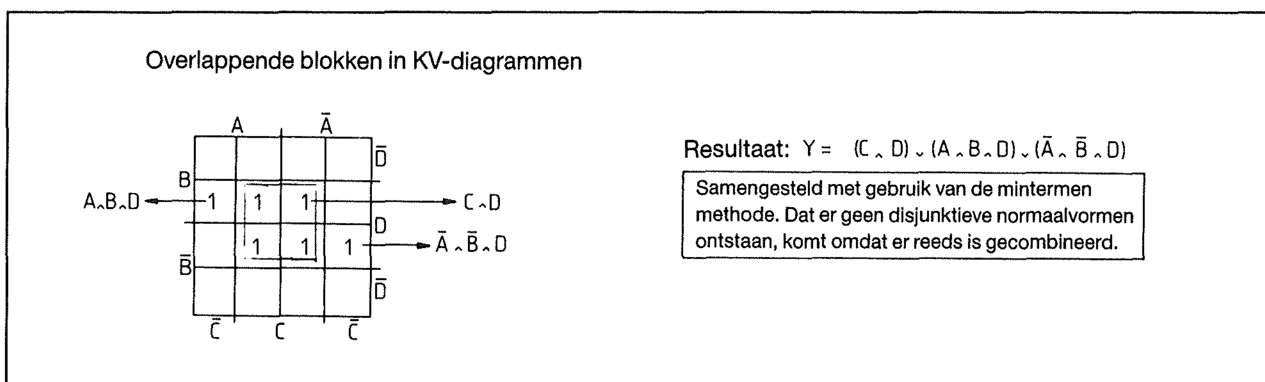
Het opstellen van de uitgangsvergelijking

Nadat men geprobeerd heeft een zo groot mogelijk blok te maken, zonder velden dubbel te gebruiken, heeft men de kleinste variant van de schakeling in het KV-diagram. Bij het uitlezen van het KV-diagram krijgt nu elk blok een waarde toegewezen, die afhankelijk van de gebruikte methode (min-termen of max-termen) conjunctief of disjunctief in de uitgangsvergelijking verschijnt. Een paar voorbeelden voor het uitlezen zijn te zien in figuur 3/6.5-8. Ter verdui-

6.5 KV-diagrammen



Figuur 3/6.5-6: Minder evidente 4-velden blokken in KV-diagram.



Figuur 3/6.5-7: Overlappende blokken

delijking geven we in figuur 3/6.5-9 een aantal voorbeelden van hoe het niet moet.

Het grensgeval

Na het invullen van het KV-diagram uit de functietabel houden we een KV-diagram met 16 ingangsvariabelen over (zie figuur 3/6.5-10). Tot onze schrik moeten we vaststellen, dat het niet mogelijk is blokken samen te stellen. Er zit dus niets anders op, dan alle individuele velden volgens de min-termen methode als conjunctionen uit te lezen. Het blijkt dus niet mogelijk dit grensgeval met de KV-diagrammen te vereenvoudigen. De enige mogelijkheid tot vereenvoudiging ligt in de Booleaanse algebra, waarin het mogelijk is enkele ingangsvariabelen buiten haken te halen. Men kan zich echter ook de moeite besparen en de meermaals voorkomende combinaties, die klaarblijkelijk in de schakeling voorkomen gewoon meerdere malen te gebruiken.

De driedimensionale KV-diagrammen

Als een KV-diagram te groot wordt, wordt het invullen van de ingangsvariabelen op de rand steeds moeilijker, omdat erop moet worden gelet, dat van veld naar veld er steeds maar een ingangsvariabele wijzigt. Bovendien is

6.5 KV-diagrammen

hergebruik van een ingangsvariabelen niet toegestaan en ook verboden combinaties, die niet zijn gedefiniëerd mogen niet voorkomen. Met zes ingangsvariabelen wordt het al een klus, waarbij het voorhoofd niet droog blijft.
Is dat het einde van het gebruik van KV-diagrammen? Als dat zo was hadden we ons de uitleg wel kunnen besparen. De oplossing ligt in het toevoegen van een extra dimensie. Hierbij creëren we een ruimtelijk KV-diagram. U moet het zien als een paar schaakborden, die boven elkaar zijn opgehangen. In figuur 3/6.5-11 is één en ander aanschouwelijk gemaakt. Zoals te zien hebben de boven elkaar liggende diagrammen dezelfde variabelen op de rand staan. De tafels moeten zo worden getekend, dat het lijkt of de velden loodrecht boven elkaar hangen.

Het invullen van het diagram is een kwestie van opletten en oefenen. Bij het vormen van blokken kunnen enerzijds velden met gelijke inhoud, die in het zelfde vlak liggen worden genomen, alsook velden, die in het vertikale vlak boven elkaar liggen. De uitgangsvergelijking wordt voor elk blok bepaald. Het principe blijft gelijk. Een KV-diagram met de complexiteit van figuur 3/6.5-11 zal de hobbyist zelden tegenkomen. We hebben het gegeven, om te laten zien, dat er geen limiet aan het principe van KV-diagrammen is. Het KV-diagram is de comfortabelste weg naar de kleinst mogelijke schakeling.

Wie de weg van de minimalisering te ingewikkeld is, of wie het teveel werk vindt kan natuurlijk uitgaande van standaard componenten een schakeling uitdenken, door stap voor stap van ingang naar uitgang te werken. De schakeling zal zelden de kleinste variant opleveren.
Als de schakeling wat ingewikkelder wordt, gaat al snel het overzicht verloren. Wat men door zich niet te verdiepen in minimalisering van de schakeling aan tijd wint bij het ontwerp, zal men dikwijls weer kwijt zijn bij het ontwerp van de (ingewikkelder) print. In figuur 3/6.5-12 geven we nog een KV-diagram formulier.

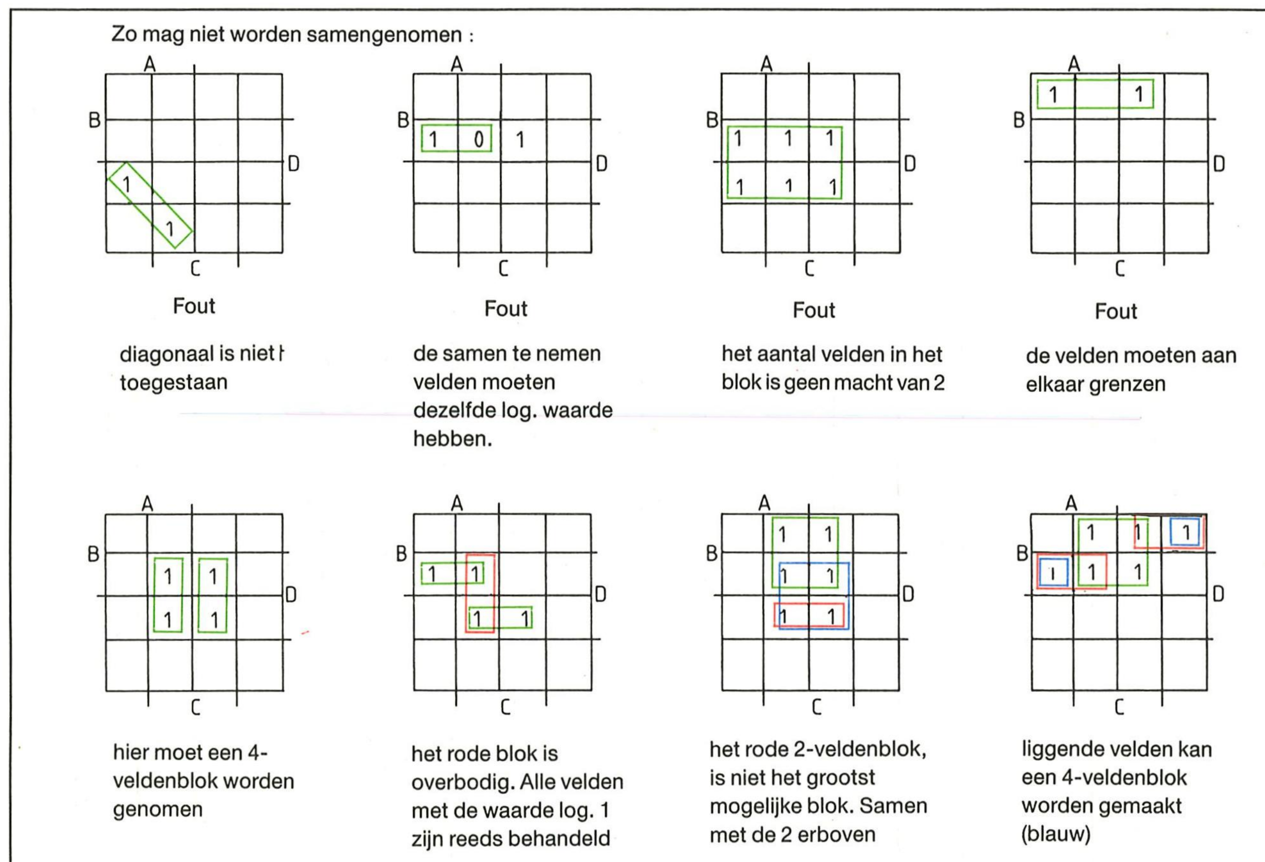
6.5 KV-diagrammen

Uitlezen van de blokvergelijkingen

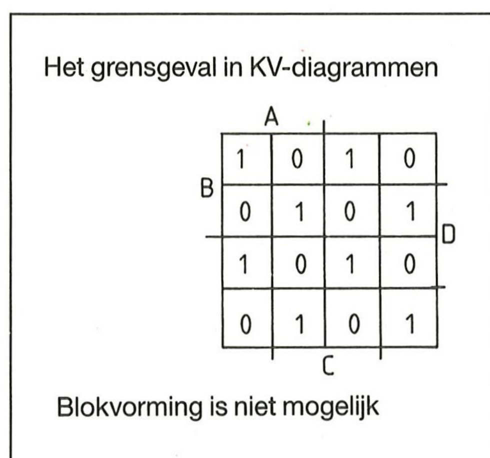
2-veldenblok	4-veldenblok	2-veldenblok	2-veldenblok 4-veldenblok	omlijnde blokken
minterm	minterm	maxterm	minterm	Uitleesmanier
$\bar{A} \rightarrow A$	rood: $A \rightarrow \bar{A}$, $C \rightarrow \bar{C}$ groen: $A \rightarrow \bar{A}$, $D \rightarrow \bar{D}$	rood: $B \rightarrow \bar{B}$ groen: $C \rightarrow \bar{C}$	rood: $A \rightarrow \bar{A}$, $C \rightarrow \bar{C}$ groen: $B \rightarrow \bar{B}$, $C \rightarrow \bar{C}$ grijs: $A \rightarrow \bar{A}$, $B \rightarrow \bar{B}$ blauw: $\bar{C} \rightarrow C$	wegvallende variabelen
$B \wedge \bar{C} \wedge D$	rood: $\bar{B} \wedge D$ groen: $\bar{B} \wedge C$	rood: $A \vee B \vee \bar{D}$ groen: $\bar{A} \vee C \vee \bar{D}$	rood: $\bar{B} \wedge D$ groen: $\bar{A} \wedge D$ grijs: $C \wedge D$ blauw: $A \wedge B \wedge \bar{D}$	Uitlezen van de blok-vergelijkingen
$Y = B \wedge \bar{C} \wedge D$	$Y = (\bar{B} \wedge D) \vee (\bar{B} \wedge C)$	$Y = (A \vee B \vee \bar{D}) \wedge (\bar{A} \vee C \vee \bar{D})$	$Y = (\bar{B} \wedge D) \vee (\bar{A} \wedge D) \vee (C \wedge D) \vee (A \wedge B \wedge \bar{D})$	Eindvergelijking
		De aanduiding op de rand moet bij het uitlezen worden geïnverteerd.	Het rode blok valt weg	Opmerking

Figuur 3/6.5-8: Voorbeelden voor het uitlezen.

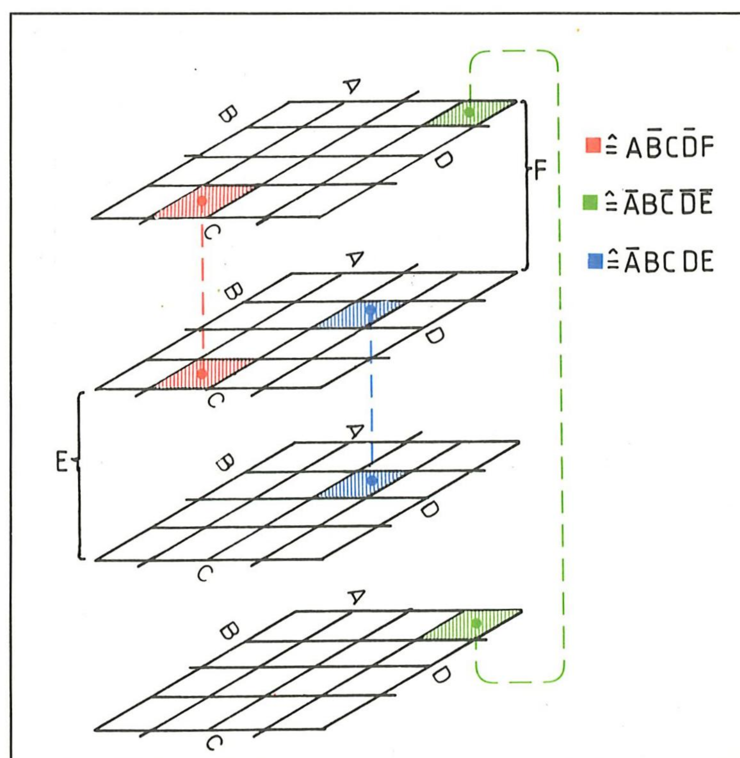
6.5 KV-diagrammen



Figuur 3/6.5-9: Voorbeelden van hoe het *niet* moet.

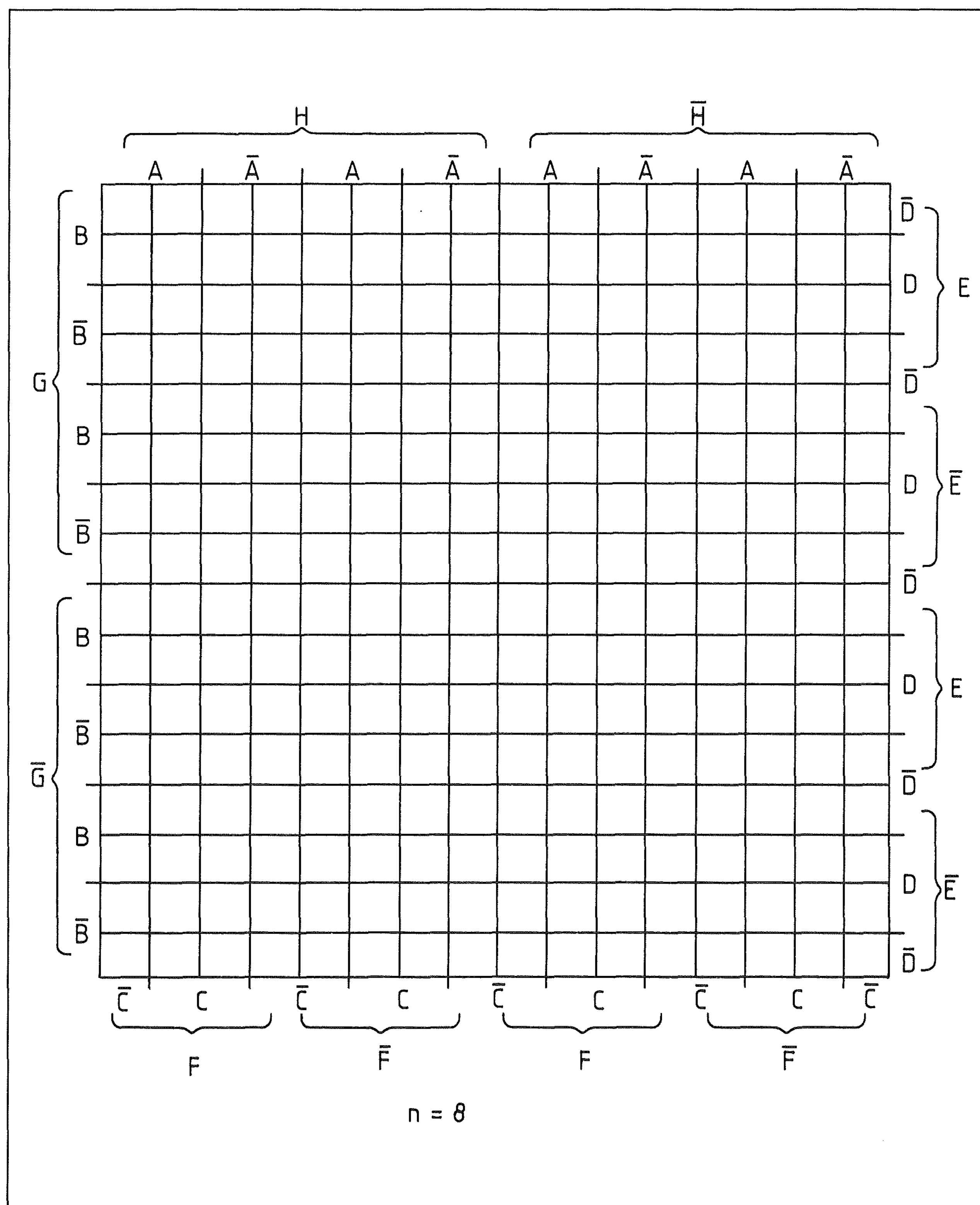


Figuur 3/6.5-10: Het grensgeval.



Figuur 3/6.5-11:
Ruimtelijk voorstelling van een KV-diagram voor 6 variabelen.

6.5 KV-diagrammen



Figuur 12: Een KV-diagram formulier.

3/6.6

De nieuwe symbolen voor digitale logika

Door de toename van de integrale dichtheid van IC's en de daarmee toenemende complexiteit van de in een schakeling verwezenlijkte functies, maken het gebruik van de oude logika symbolen te ingewikkeld om op een eenvoudige en eenduidige wijze de functie van de schakeling weer te geven.

De internationale elektronische commissie (IEC) heef daarom een nieuwe symbolische voorstelling het licht doen zien, waarbij de functie van elke uitgang ten opzichte van de ingangen duidelijk is weergegeven, zonder de interne elementen van de schakeling zelf te tonen. De nieuwe symbolen zijn meer dan alleen maar een vervanging van de oude. Het doel van de nieuwe symbolen is natuurlijk het weergeven van de grondfunctie van de schakeling. Gelijktijdig wordt echter gepoogd meer eenheid te brengen in de symbolen.

De oorspronkelijke publicaties van 1972 werden door het IEEE-committee verder ontwikkeld en onder de aanduiding "IEEE std 91-1984" in de Verenigde staten gepubliceerd. Het toenemende gebruik in databoeken, dwingt de gebruiker van digitale IC's, ook de daarin gebruikte symbolen te gaan toepassen. De volgende uiteenzetting dient om begrip te kweken voor deze symbolen en de verschillende varianten, alsmede om de overeenkomst met de huidige symbolen te verduidelijken.

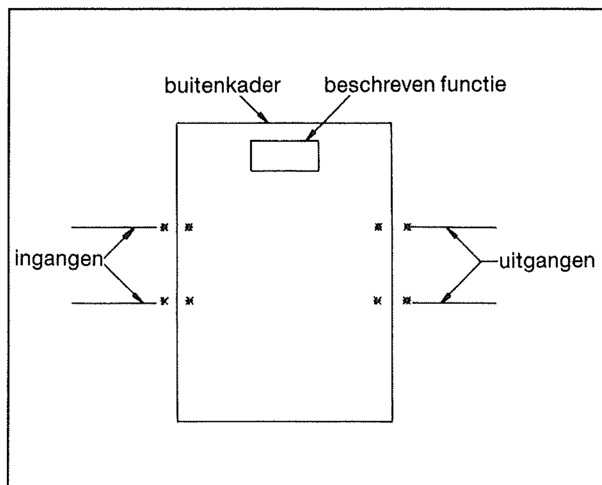
De vormgeving van de symbolen

In tegenstelling tot de oude symbolen, omvatten de nieuwe de gehele logische functie, d.w.z. dat de gehele functie met een rechthoek is omkaderd. De functie van de schakeling wordt in het bovenste derde deel van de rechthoek aangeduid. Tabel 3/6.6-1 geeft een overzicht van de gebruikte codes en afkortingen. Hoe zo'n nieuw symbool er in zijn simpelste vorm uitziet, kunt u zien in figuur 3/6.6-1. De richting van de signalen door het symbool, is normalerwijze van ingang naar uitgang (overeenkomend met de richting van links naar rechts). Mocht de richting afwijken, dan wordt dit afzonderlijk aangegeven.

Als een ingangssignaal in meerdere elementen van de functie wordt gebruikt, of als stuurfunctie fungeert (denk bijv. een "enable" signaal), dan wordt de ingang niet bij elk element aangegeven, maar een apart controleblok getekend. (zie fig. 3/6.6-2)

Het gemeenschappelijke controleblokje wordt met een inkeping van het hoofdblok gescheiden. Dit is de enige toegelaten afwijking van het rechthoekig kader.

6.6 Nieuwe symbolen digitale logika



Figuur 3/6.6-1: Algemene indeling van de symbolen

Het ingangssignaal “a” heeft invloed op alle eronder liggende elementen. Op welke wijze dit gebeurt is afhankelijk van de toegepaste functie, die dan ook altijd wordt aangegeven.

Als de schakeling een gemeenschappelijk uitgangselement bevat, dan wordt dit getekend op de manier, die u in figuur 3/6.6-3 kunt vinden. Een gemeenschappelijk uitgangselemenet kan de logische verbinding van ongeacht welk aantal ingangen zijn. Om dit op een eenduidige manier aan te geven, is het nodig binnen het element een verdere verduidelijking op te nemen. Hierover later meer.

Aanduiding van ingangen en uitgangen

De aanduiding van in- en uitgangen moet in twee groepen worden verdeeld. Eerst komen de symbolen aan de beurt, die de onderlinge verhouding van de in- en uitgangen aangeven in afhankelijkheid van het optreden van logische signalen. De tweede groep komt later aan de beurt. Grotendeels werden de aanduidingen van de oude symbolen gebruikt. Ze zijn er alleen anders uit gaan zien. Zie tabel 3/6.6-2.

Het valt op, dat de aanduiding tamelijk algemeen is gehouden. In de oude aanduiding was het zo, dat de invertering van een ingang aangaf, dat een aangelegd 0 niveau intern als een logisch 1 niveau werd verwerkt. Het logisch 1 niveau kenmerkt op actieve toestand. In de oude symbolen is de interpretatie afhankelijk van de gebruikte logika (positief of negatief). De nieuwe symbolen geven een algemene aanduiding, die zowel in positieve, als in negatieve logika werkt.

Symbolen binnen het kader

Dit is de tweede groep aanduidingen. De symbolen van tabel 3/6.6-3 worden binnen het kader gebruikt om een bepaalde hardware toestand aan te duiden (bijv. open collector). Een dergelijke aanduiding bestond in de oude symbolen niet, er was steeds tekst nodig, om de hardware situatie te definiëren. De meeste symbolen spreken voor zichzelf.

Buiten de hier gegeven symbolen zijn er nog een aantal zelden voorkomende symbolen in gebruik in (meestal) computer schakelingen. Als er binnen het kader een niet standaard symbool wordt gebruikt, is dit symbool doorgaans tussen vierkante haken geplaatst. Dergelijke symbolen worden soms door fabrikanten gebruikt.

Aanduiding van de functie

In de functieaanduiding ligt de kracht van de nieuwe symbolen.

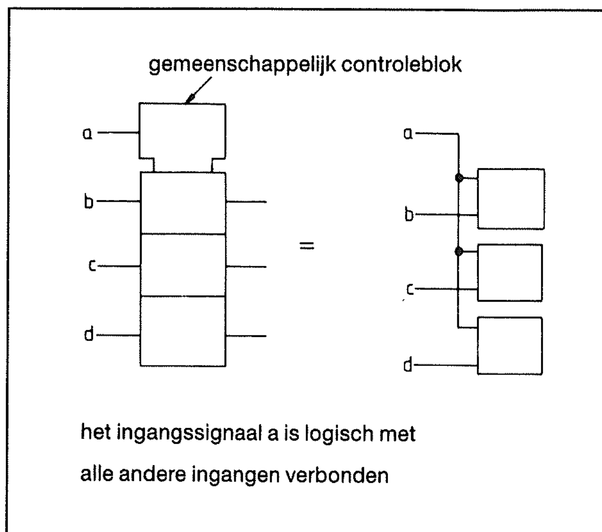
6.6 Nieuwe symbolen digitale logika

symbool	omschrijving	voorbeeld
&	AND-poort of functie	7400
≥1	OR-poort of functie	7402
=1	EXOR-poort of functie	7486
=	EXNOR-poort of functie	74180
2k	een even aantal ingangen moet actief zijn	74180
2k+1	een oneven aantal ingangen moet actief zijn	74ALS86
1	de ingang is actief bij log. 1	7404
▷ oder ◁	een drijver of element een verhoogd uitgangsvermogen (de oriëntatie van het symbool is afhankelijk van de signaalrichting)	74S436
⎍	Schmitt-trigger; element met hysteresis	74LS18
X/Y	code converter (bijv. BCD/binair; BCD/7-segment; enz.)	74LS37
MUX	multiplexer/data-selector	74150
DMUX oder DX	demultiplexer	74138
Σ	optelschakeling (Adder)	74LS385
P-Q	aftrekschakeling (Subtractor)	74LS385
CPG	snelle Carry generator	74182
π	multiplier	74LS3841
COMP	comparator	74LS682
ALU	arithemetic logic unit	74LS381
⎍	hertriggerbare monostabiele schakeling	74422
1 ⎍	niet hertriggerbare monostabiele schakeling (non-shot)	74121
G ⎍⎍	astabile schakeling met getoonde golfvorm	74320
!G ⎍⎍	synchrone astabile schakeling	74LS624
G! ⎍⎍	astabiele schakeling, die na een complete puls stopt	*
SRGm	schuifregister; m=bitlengte	74LS595
CTRm	teller; m= aantal bits, c.q. cyclus-lengte 2^m	74LS590
CTR DIVm	teller met cyclus-lengte m	74LS668
RCTRm	asynchrone teller met cyclus-lengte m	*
ROM	lees-geheugen (Read only memory)	74187
RAM	vrij toegankelijk geheugen (Random Acces memory)	74170
FIFO	vertragingslijn (first in - first out)	74LS222
I=O	schakeling te wissen met log. Ø	74AS877
I=1	schakeling te activeren met log. 1	74LS608
ϕ	met beperkte functie aanduiding (hiervoor zijn speciale regels nodig)	*

* Deze symbolen worden zelden gebruikt

Tabel 3/6.6-1: Gebruikte aanduidingen en afkortingen

6.6 Nieuwe symbolen digitale logika



Figuur 3/6.6-2: Het gemeenschappelijke controleblok

Van de gebruiker wordt echter wel een zeker omschakeling verwacht. Van de oude symbolen was bekend, dat er een vaste relatie bestond tussen de functie en de hardware. De schakeltechnische relatie is echer uit de nieuwe symbolen verdwenen. Het is dus niet meer mogelijk uit de symbolen het aantal gebruikte poorten te bepalen. Het enige dat nog telt is de functie.

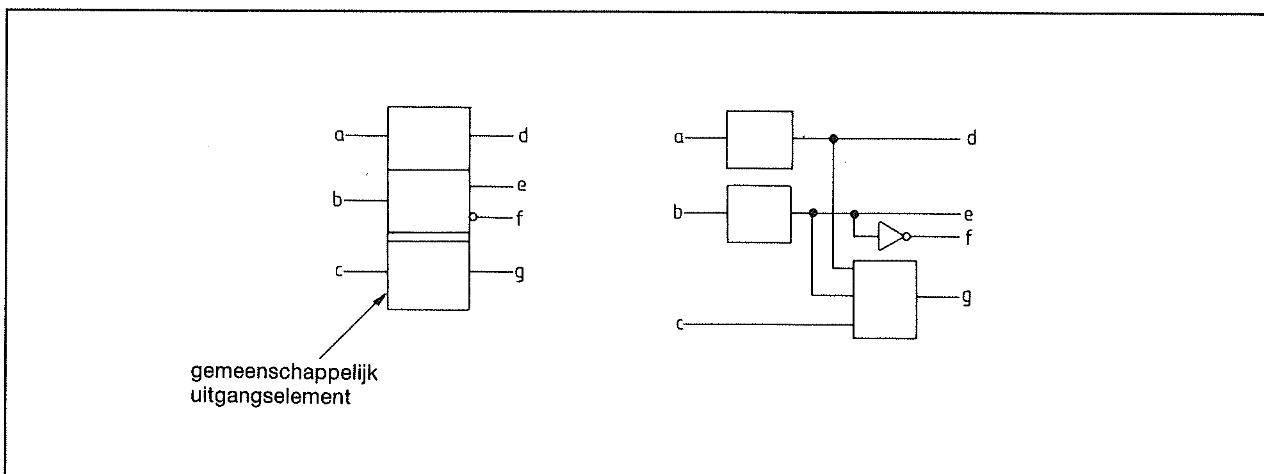
Door de IEC-commissie werden elf letters gedefiniëerd, om de functie aan te duiden. De gebruikte letters hebben de volgende betekenis:

G AND-functie of verbinding
V OR-functie of verbinding
N Inverter (EXOR-functie of verbinding)
Z Interne logika
X Overdrachts functie
C Controle functie
S Set-ingang bij bistabiele elementen
R Reset-ingang bij bistabiele elementen
EN Enable ingang
M mode
A adres

De functies zijn inmiddels zo complex, dat we de individuele functies nog eens onder de loep nemen, om ze te begrijpen.

G-functie

De G-functie is de AND verbinding (conjunctie) van de optredende ingangsvariabelen. Tot nu toe waren er voor conjunctie talloze verschillende



Figuur 3/6.6-3: Het gemeenschappelijke uitgangselement

6.6 Nieuwe symbolen digitale logika

log. ingangsinversie. Een externe 0 levert intern 1
log. uitgangsinversie. Een interne 1 levert extern 0
low-actif ingang (≘ —◁ in positieve logika)
low-actif uitgang (≘ ▷— in positieve logika)
low-actif uitgang, met signaalrichting van rechts naar links
signaalrichting van rechts naar links (zonder aanduiding altijd van links naar rechts)
bidirectionele signaalrichting

dynamische ingangen als volgt te activeren	positieve logika	negatieve logika	benodigde pulsflank
	1 → 0 (niet gebruikt)	1 → 0 (niet gebruikt)	niet gebruikt (H → L)
	0 → 1	0 → 1	L → H

niet-log verbinding, gewoonlijk gevolgd door een nader aanduiding in het symbool
analoge ingang in een digitaal circuit
digitale ingang aan een analoog circuit
interne verbinding (zonder invertering)
interne verbinding (met invertering)
dynamische interne verbinding
Een pos. flank op de ingang zorgt voor een log. 1
interne (virtuele) ingang; is altijd log. 1, totdat een hogere afhankelijkheid de ingang beïnvloed
interne (virtuele) uitgang; wordt door een interne ingang, waarvan de functie nader is omschreven beïnvloed

Tabel 3/6.6-2: Aanduiding van in- en uitgangen

aanduidingen in zwang. Nu is alleen nog de G-functie toegelaten. In tabel 3/6.6-4 vindt u enige voorbeelden. Het valt op, dat hier achter de letter “G” een getal (hier een 1) staat. Dit geeft aan, dat de beide ingangen met hetzelfde getal conjunctief verbonden zijn.

Als een ontkenning moet worden aangeduid, dan verschijnt er boven het getal een liggend streepje. Het getal kan ook worden vervangen door een letter uit het Griekse alfabet. De conjunctieve verbinding is natuurlijk ook mogelijk met dynamisch (flankgestuurde ingangen).

V-functie

De letter V is de aanduiding voor een OR-verbinding (disjunctie), zie wederom tabel 3/6.6-4. Als een V-ingang logisch 1 is, dan zijn alle afhankelijke uitgangen ook logisch 1. Bij een logische 0 is de toestand van de logische uitgangen afhankelijk van eventuele andere ingangen, waarvan de betreffende uitgang afhankelijk is.

N-functie

De inversie van een ingang kan een EXOR—functie vervullen, als deze schakelbaar is.

Ook dit is in tabel 3/6.6-4 terug te vin-

6.6 Nieuwe symbolen digitale logika

	Ondergeschikte uitgang (van een flank-getriggerde flip-flop). De uitdaging verandert, als de bepalende ingangsflank (bijv. de klok) terug komt in de stabiele toestand
	Ingang met hysteresis
	Open collector uitgang van een NPN transistor, die een relatief lage uitgangs impedantie kan sturen. Er is een pull-up weerstand naar de pos. voeding nodig. Kan ook worden toegepast in wired-AND logika.
	Passieve pull-up uitgang. De open collector is intern reeds van een pull-up weerstand voorzien.
	Open emitter uitgang van een NPN transistor, die een logische 1 kan leveren in een relatief lage impedantie. Heeft een pull-down weerstand naar aarde nodig. Kan ook worden toegepast in wired-OR Logika.
	Passieve pull-down uitgang. De open emittor is intern reeds van een pull-down weerstand voorzien.
	Tri-state uitgang.
	Uitgang met verhoogde fan-out (Het symbool geeft de signaal richting aan)
EN	Enable ingang bij logische 1 zijn alle uitgangen actief bij logische 0 zijn alle open collector cq emittor uitgangen niet geleidend, alle tri-state uitgangen in de hoogohmige toestand en alle andere uitgangen (bijv. totem-poles) logisch 0.
J,K,R,S,T	Wordt meestal bij flip-flops gebruikt (bijv. R=Reset, T=Toggle)
D	Data ingang van een geheugen element (komt overeen met S R)
→m ←m	naar rechts (links) schuivend; m=aantal ingangen } m=1 wordt doorgaans weggelaten
+ m - m	voorwaarts (achteruit) tellen; m=aantal ingangen
0 ... m }	Binaire groepering, m is het meest significante bit.
CT = 15	Als de ingang actief is, zal de uitgang de aangegeven waarde aannemen
CT = 9	Als het register de aangeduide waarde bereikt, wordt de uitgang actief.
	Ingangs groepering, geeft aan, dat meerdere ingange samen een logische poort vormen bijv. de uitbreidings ingangen van een 7450. X $\overline{X}$ E
"1"	Vaste logische toestand van de uitgang.

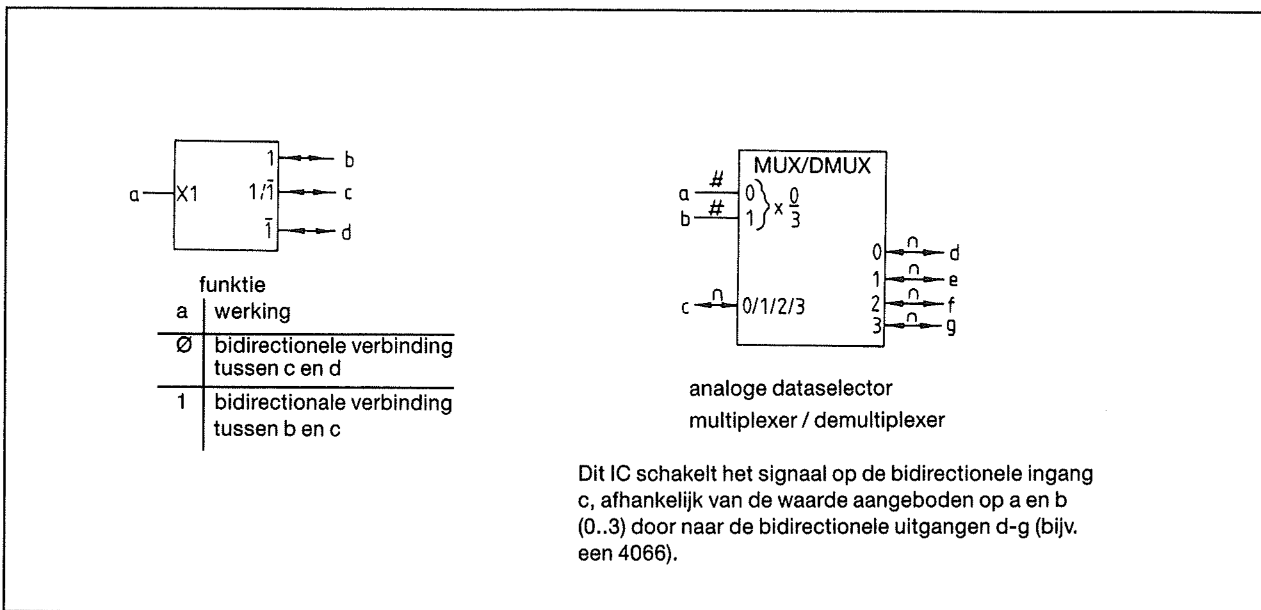
Tabel 3/6.6-3: Gebruikte symbolen binnen het kader

6.6 Nieuwe symbolen digitale logika

functie	voorbeeld	komt overeen met
G AND		
V OR		
N negatie exclusieve-OR		gestuurde inversie b = 0 , dann c = a b = 1 , dann c = $\bar{a}$
Z interne logika		

Tabel 3/6.6-4: De verschillende verbindingen (functies)

6.6 Nieuwe symbolen digitale logika



Figuur 3/6.6-4: De X-functie (overdracht)

den. Inversie van a is afhankelijk van de logische toestand van b. Zonder b hebben we met een recht toe recht aan inversie te maken.

Z-functie

Een interne verbinding tussen in- en uitgangen wordt met de letter Z aangeduid. Tussen in- en uitgangen wordt een functie vervult, die verder niet van belang is. (Bijvoorbeeld een buffer trap van een uitgang). Zie ook tabel 3/6.6-4.

X-functie

Een X-functie is een aanduiding van een bidirectionele afhankelijkheid van in- en uitgang. Het gaat daarbij om de overdracht van een analoog signaal aan digitale schakelingen.
De richting waarin het signaal loopt is afhankelijk van de aangelegde potentialen, of van de functie van de schakeling. In figuur 3/6.6-4 wordt deze functie aan de hand van een voorbeeld verduidelijkt.

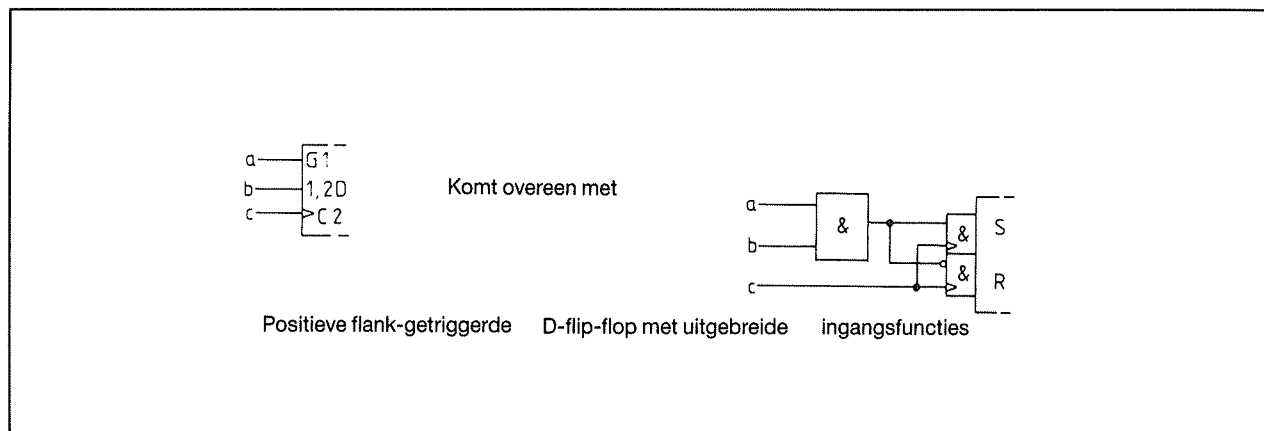
C-functie

Hiermee wordt een controle functie aangeduid. De controle functie maakt het mogelijk de reactie van de uitgangen op de overige ingangen te sperren of vrij te geven. Dit gebeurt meestal bij geheugen elementen, flip-flops e.d. (met de data ingangen wordt in dit geval bedoeld J-, K-, D-, R-, S- ingangen). Naast statische ingangen zijn er ook nog dynamische ingangen, die op de flanken (edge in het Engels) van het aangelegde signaal reageren. Zie figuur 3/6.5-5. In het symbool is de aanduiding een driehoekje. Als de "controle" ingang logisch 1 is, dan zijn de data-ingangen vrijgegeven. Bij een logische 0 zijn de data-ingangen gesperd.

S- en R-functie

Wanneer een RS-flip-flop wordt samengesteld uit NOR-poorten, dan kan er in principe een ongeorloofde toestand optreden. Deze toestand ontstaat, als beide ingangen gelijktijdig logisch 1

6.6 Nieuwe symbolen digitale logika



Figuur 3/6.6-5: C (controle)-functie

zijn. In de nieuwe symbolen, worden deze toestanden aangeduid en derhalve gedefiniëerd. Ook al definieer je de toestanden, daarmee zijn ze natuurlijk nog niet geoorloofd. De uitgangstoestand is onder deze condities niet te voorspellen.

EN-functie

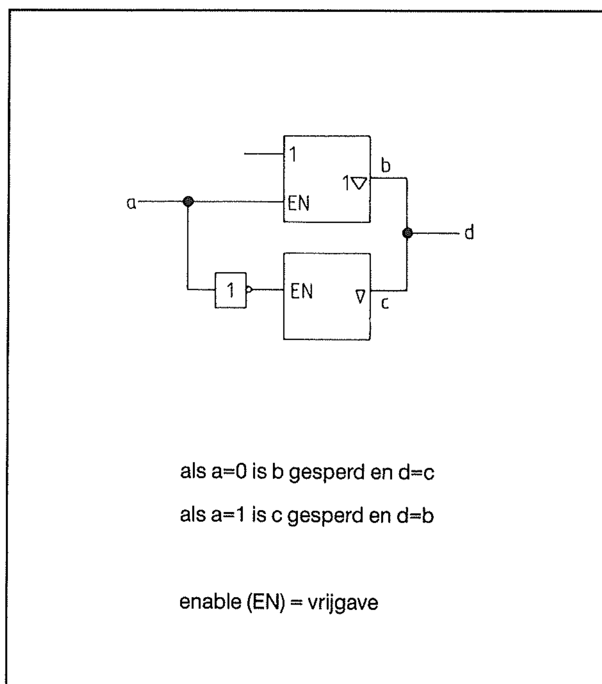
De "enable" ingang X (ENx), figuur 3/6.6-7, stuurt alleen de met x gemerkte ingangen (enable= vrijgave). In tegenstelling met een EN-ingang, die alle uitgangen gelijktijdig activeert, kan door de ENx-ingang een selectie worden aangegeven. Als op de EN-ingang een logische 1 staat, dan zijn de door de EN-ingang vrijgegeven ingangen bepalend voor de functie van de schakeling. De uitgangen hebben gedefiniëerde toestanden. Als de EN-ingang logisch 0 is, dan zijn alle door de EN-ingang ge-

S	R	Q	$\bar{Q}$	Q	$\bar{Q}$	Q	$\bar{Q}$	Q	$\bar{Q}$	Q	$\bar{Q}$
0	0	st	st	st	st	st	st	st	st	st	st
0	1	0	1	0	1	0	1	0	1	0	1
1	0	1	0	1	0	1	0	1	0	1	0
1	1	?	?	1	0	0	1	1	1	0	0

st= storend (oude toestand) ?= onbepaald

Figuur 3/6.6-6: S-(set) en R-(reset) functie

6.6 Nieuwe symbolen digitale logika



Figuur 3/6.6-7: EN (enable)-functie

stuurde in- en uitgangen gesperd. Voor open collector uitgangen betekent dit, dat de uitgangs-transistor is gesperd en voor de tri-state uitgangen betekent dit, dat de uitgang naar buiten toe hoogohmig is. Alle andere uitgangen (bijv. totem-poles) zijn intern logsich 0.

M-functie

In de nieuwe norm kan worden aangegeven, dat een schakeling op verschillende wijzen (modes) kan werken. De functie van de schakeling, is dan afhankelijk van de gekozen wijze van werken. In elektronica termen: "afhankelijk van de gekozen mode". Als een in- of uitgang in meerdere modes dezelfde functie vervult, dan wordt dit weergegeven, door het mode-nummer of nummers tussen haken en door schuine strepen gescheiden te vermelden.

M-functie voor ingangen

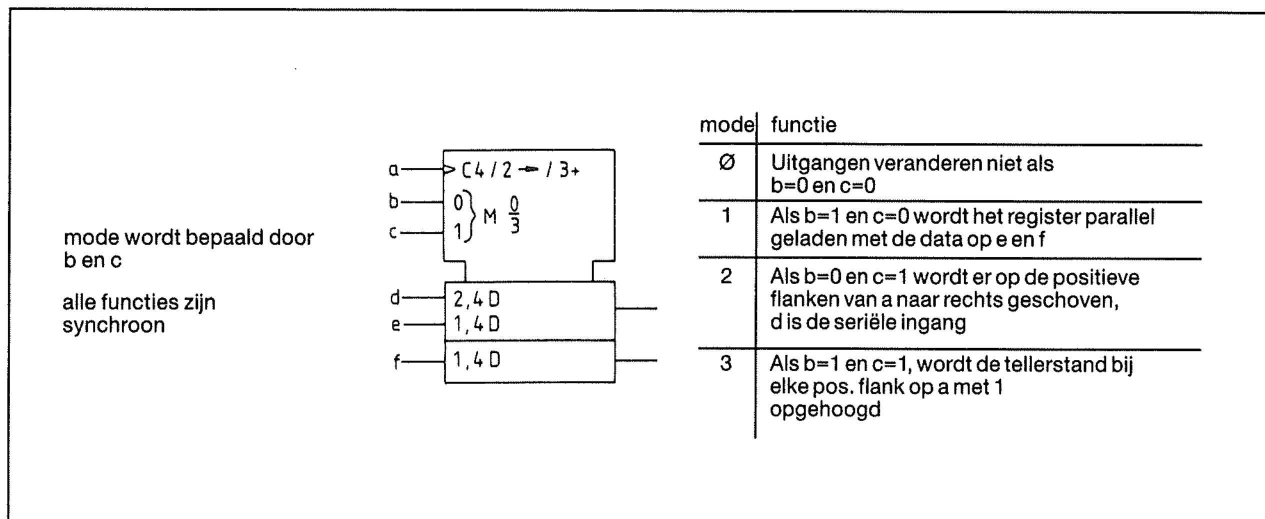
Hier komt de M-functie overeen met de C-functie, zij het, dat er meer varianten mogelijk zijn. Als een gestuurde ingang meerdere (door schuine strepen gescheiden) aanduidingen heeft (bijv. 4/2-/3+), dan is de mode voor die functie irrelevant. Het gaat dan om een universele, multifunctionele ingang. De schakeling van figuur 3/6.6-8a, heeft vier verschillende werkingswijzen (modes). Welke van deze modes actief is, wordt bepaald door de signalen op de ingangen "b" en "c". De ingangen "d", "e" en "f" zijn D-ingangen, die de parallel in te lezen data voeren, die in mode "1" wordt ingelezen. In mode "2" wordt de data in het register naar rechts geschoven door de flankgetriggerde ingang "a". In mode "3" wordt de inhoud van het register bij elke positieve flank op "a" met 1 opgehoogd.

M-functie voor uitgangen

De M-functie voor uitgangen is aanzienlijk ingewikkelder te beschrijven. De uitgang van een element is dikwijls slechts dan actief, als een bepaalde met een index nummer overeenstemmende mode is gekozen. Als een uitgang meerdere (door schuine strepen gescheiden) nummers heeft (bijv. 2,4/3,5), dan is de uitgang niet actief, als een andere niet aangeduide mode actief is.

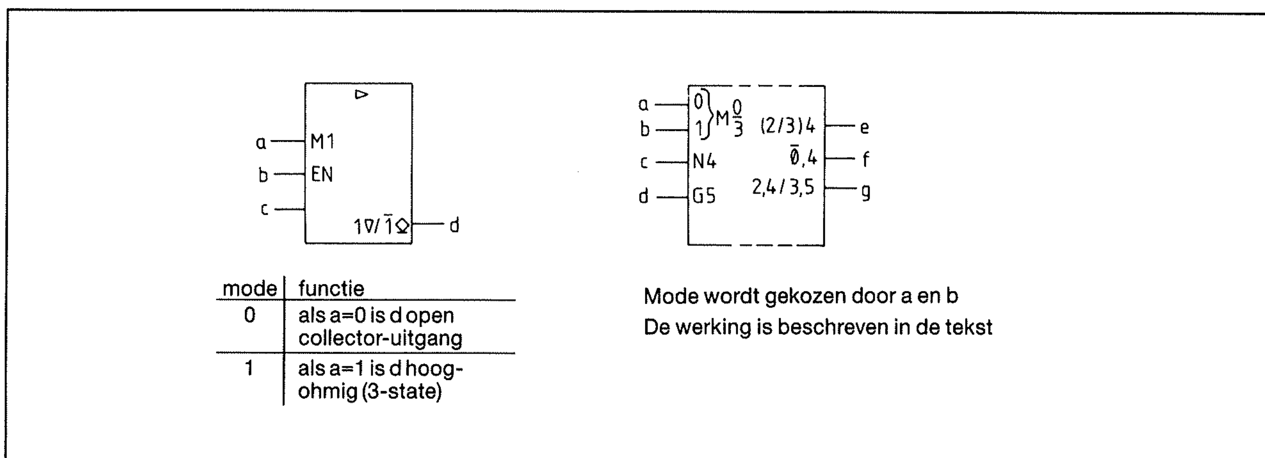
In het voorbeeld van figuur 3/6.6-8b is de uitgang "d" tri-state in mode "1" (als a=1), of open collector uitgang in mode "2" (als a=0). In figuur 3/6.6-9 is de uitgang "b" een uitgang van het gemeenschappelijke controlblok. Deze uitgang is slechts dan logisch 1, als ingang "a" = log. 1 en de inhoud van het register gelijk is aan "negen" (CT=content).

6.6 Nieuwe symbolen digitale logika



mode	functie
Ø	Uitgangen veranderen niet als b=0 en c=0
1	Als b=1 en c=0 wordt het register parallel geladen met de data op e en f
2	Als b=0 en c=1 wordt er op de positieve flanken van a naar rechts geschoven, d is de seriële ingang
3	Als b=1 en c=1, wordt de tellerstand bij elke pos. flank op a met 1 opgehoogd

Figuur 3/6.6-8a: M(mode)-functie voor ingangen



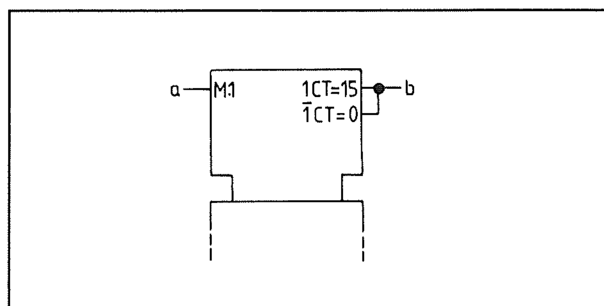
mode	functie
0	als a=0 is d open collector-uitgang
1	als a=1 is d hoog-ohmig (3-state)

Figuur 3/6.6-8b: M(mode)-functie voor uitgangen

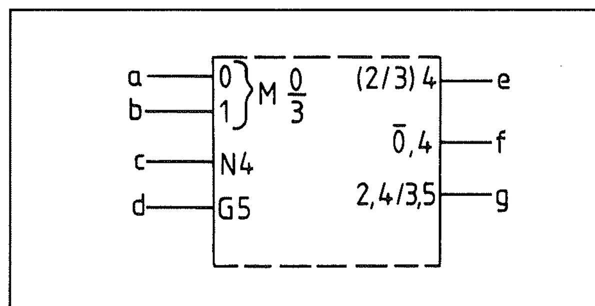
Figuur 3/6.6-10 is een voorbeeld van een variant op bovenstaande beschrijving. Als we figuur 3/6.6-11 bestuderen, zien we de onderlinge verhoudingen van de in- en uitgangen. Laten we ze eens doorlopen. De mode wordt bepaald door ingangen "a" en "b". Zij hebben bovendien een binaire waarde. In mode "2" en "3" verschijnt op uitgang "e" de inversie van ingang "c". Op uitgang "f" vinden we de inversie van ingang "c", als mode "0" niet is gekozen. Men had dit bij uitgang "f" ook weer kunnen geven met: (1/2/3)4. Uitgang "g" kan twee verschillende functies hebben. In mode "2" levert hij de inverse van ingang "c". In mode "3" is het de AND-functie van ingang "d" en uitgang "g".

We merken nog op, dat bij mode "0" alle uitgangen inactief zijn en zich dus in hun gedefiniëerde toestand bevinden. Dit geldt ook voor de op een be-

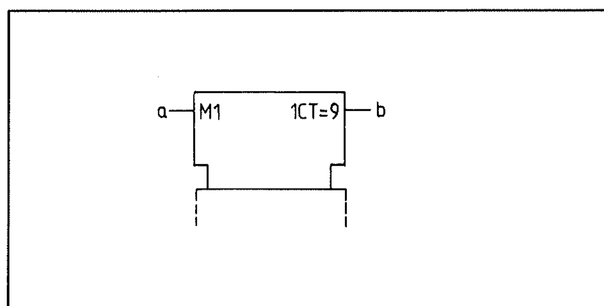
6.6 Nieuwe symbolen digitale logika



Figuur 3/6.6-9: Uitgang van een controleblok



Figuur 3/6.6-11: M-functie op uitgangen



Figuur 3/6.6-10: Modificeerbare uitgang

paald moment niet actieve uitgangen, d.w.z. voor uitgangen waarvan voor een bepaalde mode geen functie is gespecificeerd.

Het zal nu reeds duidelijk worden hoe kompakt we met deze nieuwe symbolen de functie van een schakeling kunnen voorstellen. Bij schakeltechnisch schema's met de oude symbolen, is de functie van de schakeling vaak nauwelijks of met zeer veel moeite terug te vinden.

A-functie

Om uit een array (=reeks) met meerdere dimensies een element of groep elementen (woord) aan te kunnen duiden is het noodzakelijk, dat dit element kan worden geadresseerd. De A-(=adres) functie is de symbolische aanduiding hiervan.
Afhankelijk van de logische toestanden op de A-ingangen wordt een bepaald element uit de array geselecteerd. De uitgangen van de individuele arrays zijn middels een OR-functie met elkaar verbonden. Omdat de meeste arrays zo zijn geconstrueerd, dat slechts een element tegelijkertijd actief is, spaart men op deze wijze uitgangen uit.
Verdere ingangen (bijv. klok, enable e.d.) worden onafhankelijk van de adres-ingangen toegevoegd. In de symbolische aanduiding volgt op de A altijd een getal, dat de geselecteerde sektie aanduid.

In figuur 3/6.6-12 zien we de voorstelling van een geheugen van 3 bij 2 bits. De afbeelding rechts is een verder uitgewerkt voorbeeld van dezelfde functie. Door de enable signalen EN1 tot EN3 worden telkens de D-flips-flops vrijgegeven. Het inlezen van data gebeurt middels klokingang "d". De OR-functie op de uitgang zorgt ervoor, dat hierop de data verschijnt van het geselecteerde geheugenelement. Samen vormen de uitgangen "f" en "h" de binaire uitgangswaarde van het geheugen. Als het aantal adres-ingangen groter wordt, of de adres-ingangen in blokken met enable-lijnen zijn ondergebracht dan kan voor de letter A nog een getal staan, dat de samenhang verduidelijkt,

6.6 Nieuwe symbolen digitale logika



Figuur 3/6.6-12: A-(adres) functie voorstelling van een 3x2 bit geheugen met de nieuwe norm

bijv. 1A, 2A enz.

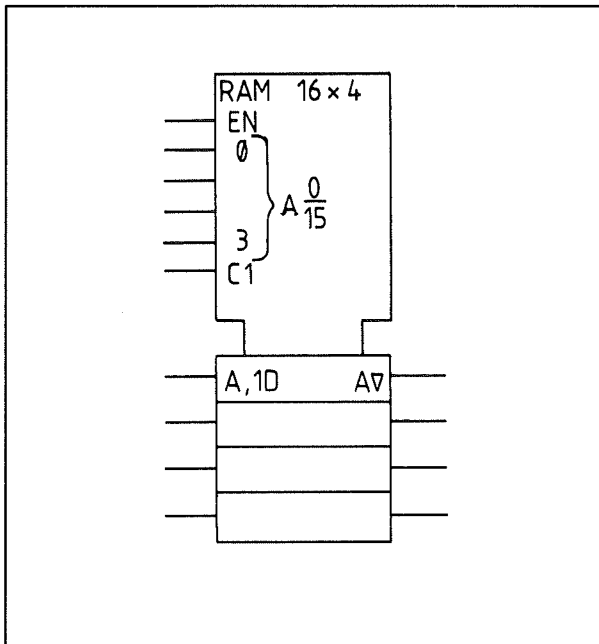
In figuur 3/6.6-13 is nog een ander voorbeeld te zien, van wat hier wordt besproken. Als men naar de uitgang van de afzonderlijke elementen kijkt, dan valt op, dat de hardware-matige opbouw slechts eenmaal te zien is. Daaruit kunt u terecht de conclusie trekken, dat alle andere elementen (geheugencellen) er evenzo uitzien.

Bistabiele elementen

De veelheid van bistabiele elementen, die men in de elektronica vindt moeten op een of andere wijze eenduidig worden weergegeven. In de nieuwe symbolen zijn ze inderdaad alle terug te vinden. De eenvoud van de symbolische voorstelling is de sleutel tot de compacte weergave, figuur 3/6.6-14.
Eenvoudige geheugen elementen hebben een niveau sturing. Zo is bijv. een D-ingang actief zolang de C-ingang (klok) op een logisch 1 niveau is. De uitgangen volgen dan de ingangen. Niveau gestuurde flip-flops accepteren data van de J-, K-, S- of R-ingangen, zolang de klok actief is. De Engelse term, die u voor "niveau gestuurde flip-flops" in databoeken vindt is: "level triggered flip-flop". Gedurende deze tijd is de toestand van de flip-flop niet gedefiniëerd en zijn storingen mogelijk. Pulsgestuurde flip-flops vereisen dat de ingangsdata stabiel is en blijft vlak voor en gedurende de klokpuls. Deze flip-flops zijn slechts gedurende een korte periode actief. De uitgang volgt de ingang als de klok weer op logisch 0 niveau arriveert.

Er zijn ook flip-flops met een soort data-lock-out. Hierbij worden de ingangen direct nadat de klokpuls de data in de flip-flop heeft ingelezen geblokkeerd.
De data op de ingangen behoeft hier slechts een korte tijd stabiel te zijn. Deze flip-flops zijn aanzienlijk storingsongevoeliger.

6.6 Nieuwe symbolen digitale logika

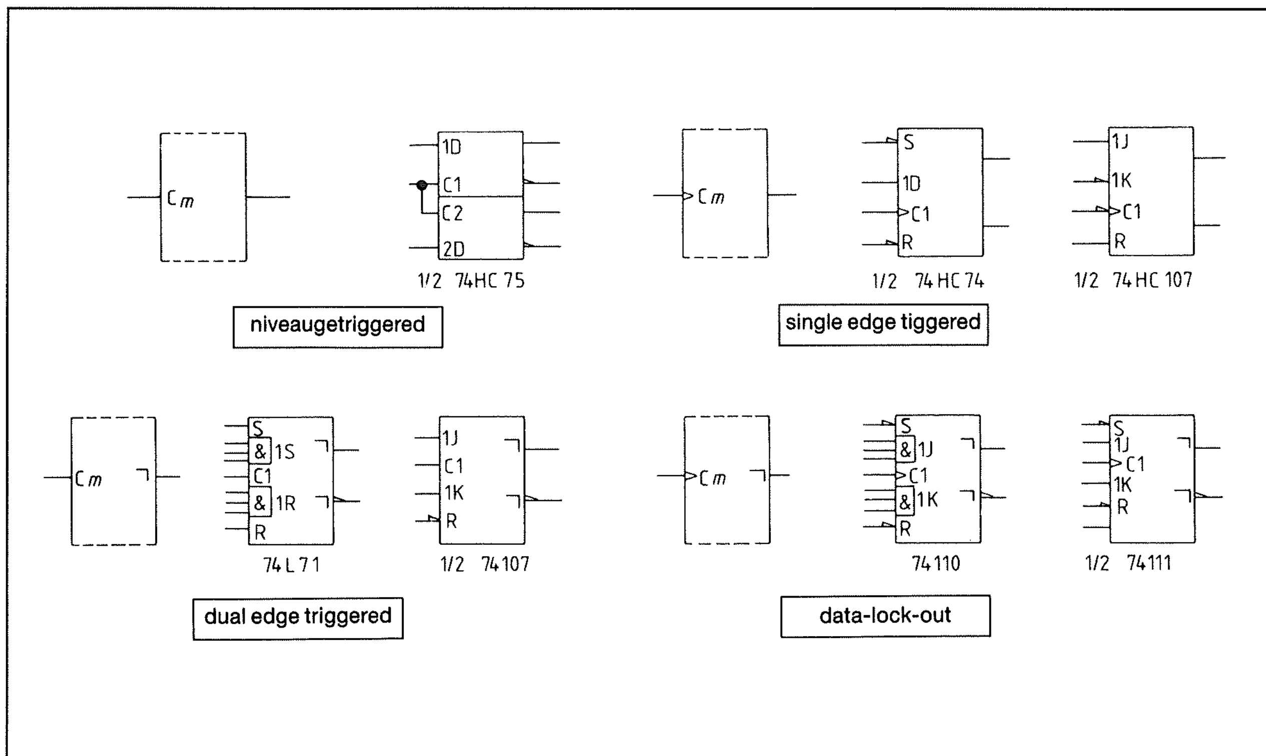


Figuur 3/6.6-13: 16x4 bit random access memory met tri-state uitgang

Opmerking: Naast de data-ingangen hebben vele geïntegreerde flip-flops asynchrone Set- en Reset ingangen, die onafhankelijk van het stuursignaal (de klok) werken. Het gelijktijdig activeren van Set- en Reset-ingangen levert meestal een instabiele uitgangstoestand op en moet dus worden vermeden.

Code-converters

Code-converters zetten een code om in een ander code, bijvoorbeeld BCD naar binair, Aiken code in exess-3, BCD naar 7-segments etc. Het algemene symbool hiervoor is figuur 3/6.6-15.
Binnen het symbool kan de code, die wordt omgezet of de code, waarnaar wordt omgezet, worden aangegeven. De interne opbouw van de schakeling bepaald de uitgangscode.



Figuur 3/6.6-14: Vier verschillende typen flip-flops volgens de nieuwe norm

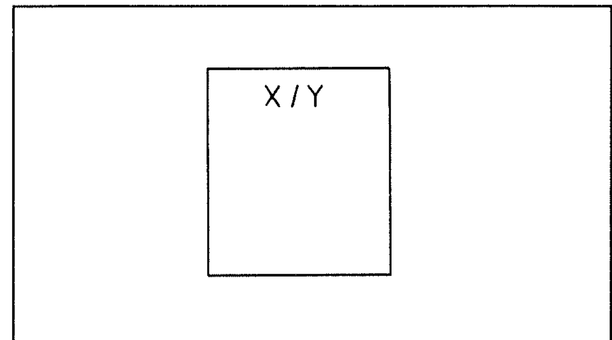
6.6 Nieuwe symbolen digitale logika

Het symbool van de PROM in figuur 3/6.6-16 geeft zoveel informatie, dat de functietabel niet nodig is.

De aanduidingen, die bij de uitgangen staan geven aan bij welke ingangswaarden de betreffende uitgang een logische 1 wordt, dan zijn alle ingangswaarden aangegeven, gescheiden door een schuine streep. Als de uitgang logisch 1 is bij een aantal opeenvolgende ingangswaarden dan kan dit worden aangegeven, door de eerste en de laatste waarde te geven gescheiden door punten (bijv. 4...9=4/5/6/7/8/9). Als uit het symbool reeds de uitgangscode duidelijk is, dan is een aanduiding als in figuur 3/6.6-7 voldoende.

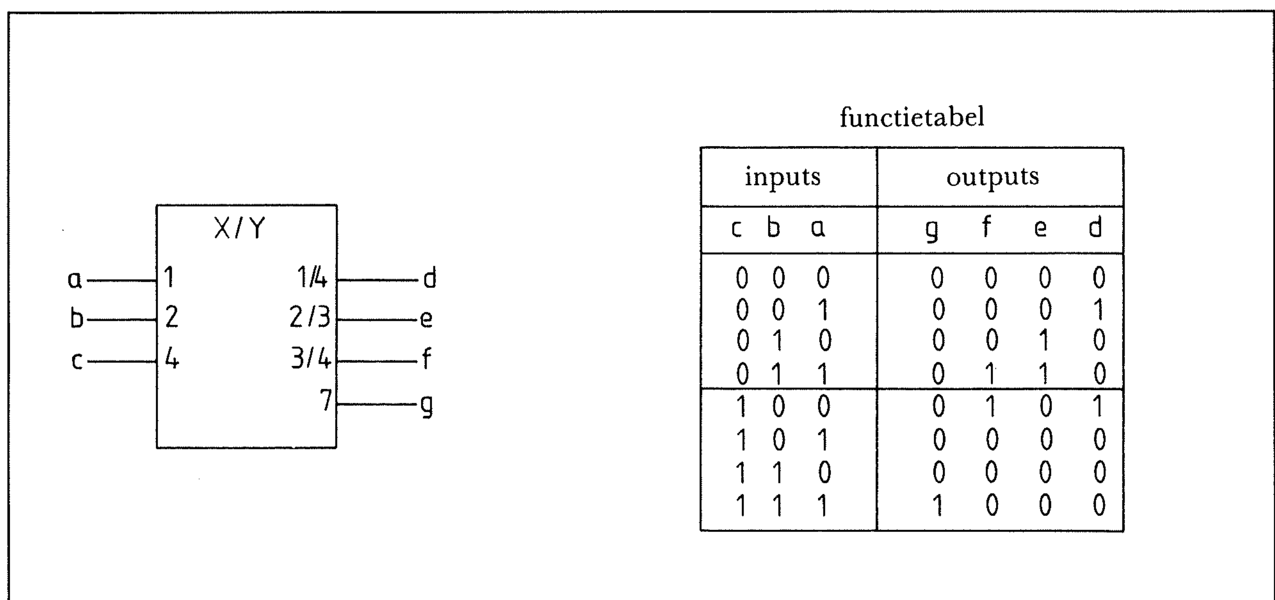
Het genereren van selectiesignalen (enables) door decoders

Decoders zijn in wezen een bijzondere vorm van code converters. Eigenlijk alleen bijzonder door de toepassing. De aangeboden ingangscode levert slechts op een uitgang een logische 1 of juist



Figuur 3/6.6-15: Algemeen code-converter symbool

een logische 0 op.
Een goed voorbeeld van toepassing vindt men in computer geheugens, waar decoders worden toegepast om bepaalde geheugen-IC's of banken van geheugen IC's te secteren. In figuur 3/6.6-17 is zo'n decoder beschreven.
Als een decoder een signaal genereert voor de keuze van verschillende modes van een erachterliggende schakeling, dan duidt men dit in het bovenste deel van het symbool aan met X/M.

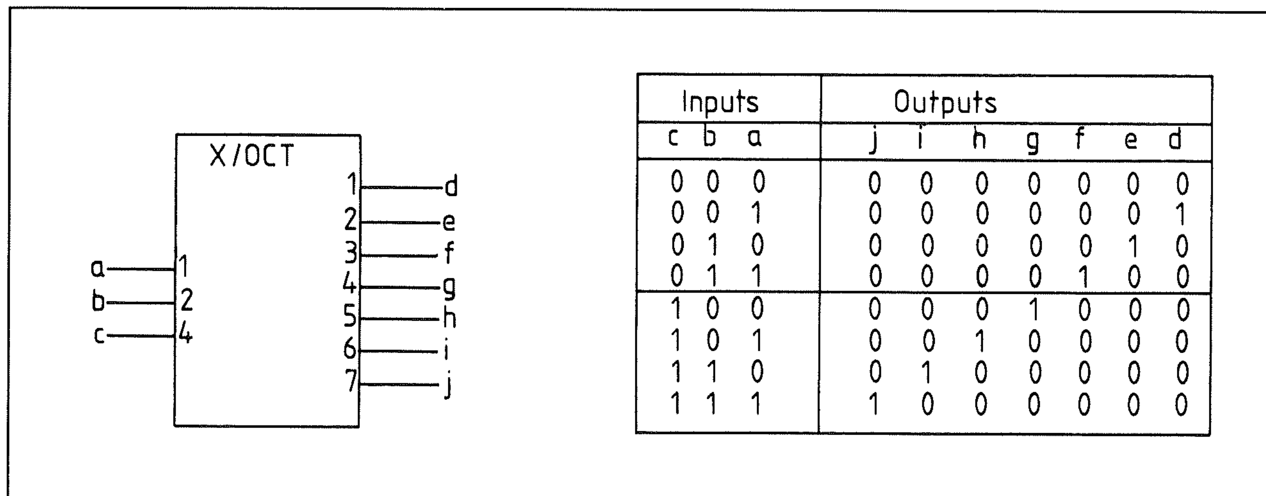


functietabel

inputs			outputs			
c	b	a	g	f	e	d
0	0	0	0	0	0	0
0	0	1	0	0	0	1
0	1	0	0	0	1	0
0	1	1	0	1	1	0
1	0	0	0	1	0	1
1	0	1	0	0	0	0
1	1	0	0	0	0	0
1	1	1	1	0	0	0

Figuur 3/6.6-16: PROM als code-converter

6.6 Nieuwe symbolen digitale logika



Inputs			Outputs						
c	b	a	j	i	h	g	f	e	d
0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0	1
0	1	0	0	0	0	0	0	1	0
0	1	1	0	0	0	0	1	0	0
1	0	0	0	0	0	1	0	0	0
1	0	1	0	0	1	0	0	0	0
1	1	0	0	1	0	0	0	0	0
1	1	1	1	0	0	0	0	0	0

Figuur 3/6.6-17: 1 uit 8 decoder

Een voorbeeld

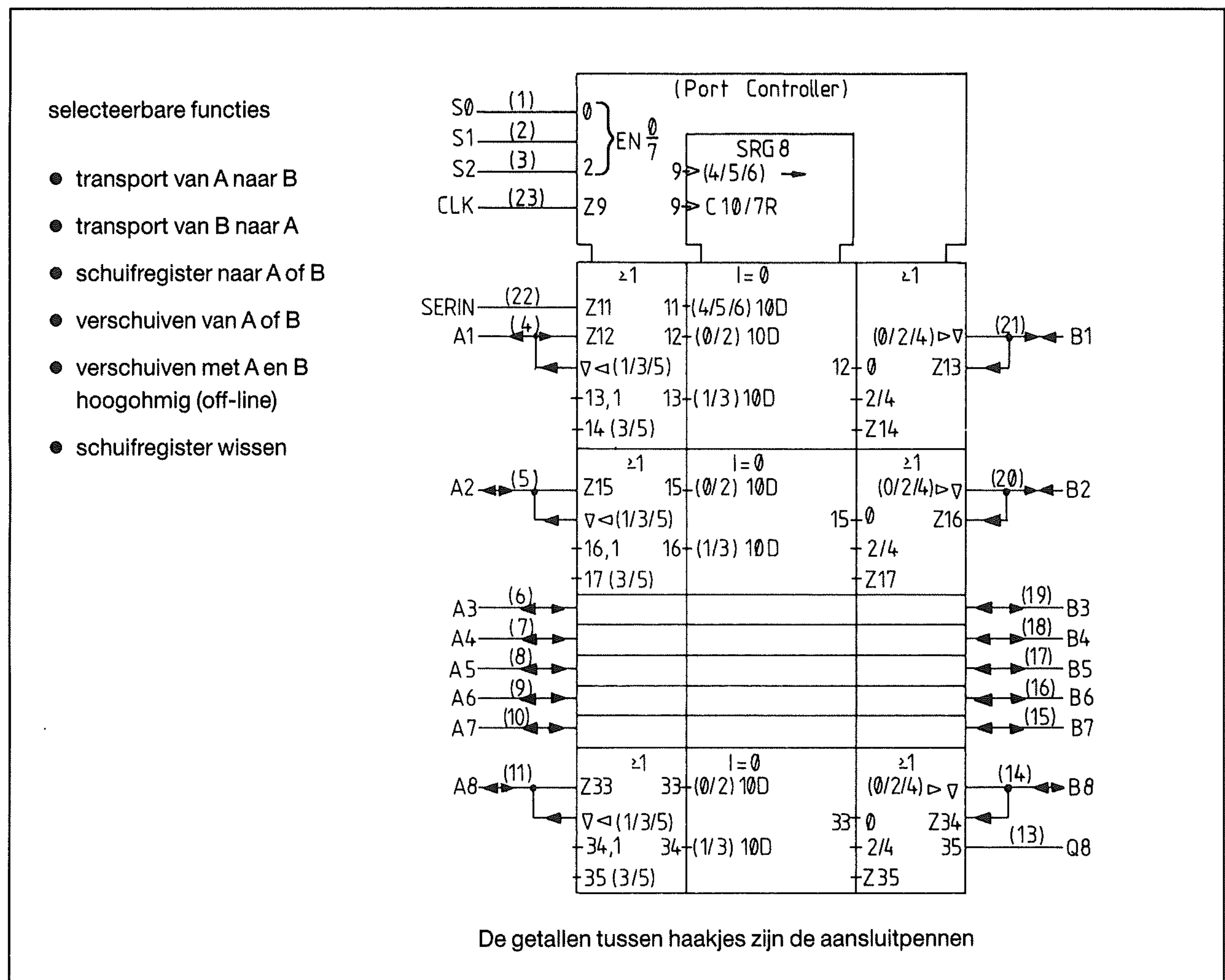
De complexiteit van de nieuwe symbolen tonen we in figuur 3/6.6-18 aan de hand van een willekeurig voorbeeld. We zien in deze figuur een complexe bus schakeling. Dit IC herbergt een groot aantal verschillende functies. Aangesloten op de bus van een microcomputer kan dit IC een seriëel naar parallel of parallel naar seriëel omzetting verzorgen. Naast het symbool zijn de functies nog eens in klare taal gegeven, om u te helpen de aanduidingen te begrijpen men de samenhang ervan te zien.

Tot slot

Naast alles wat in dit hoofdstuk aan de orde is gekomen zijn, in de normen voor de nieuwe symbolen ook nog in- en uitgangsvolgorden gedefiniëerd.
We gaan daarop niet verder in. Als u de stof uit dit hoofdstuk begrijpt, zullen deze symbolen geen problemen opleveren. Daarbij komt, dat ze in de praktijk niet of nauwelijks voorkomen.

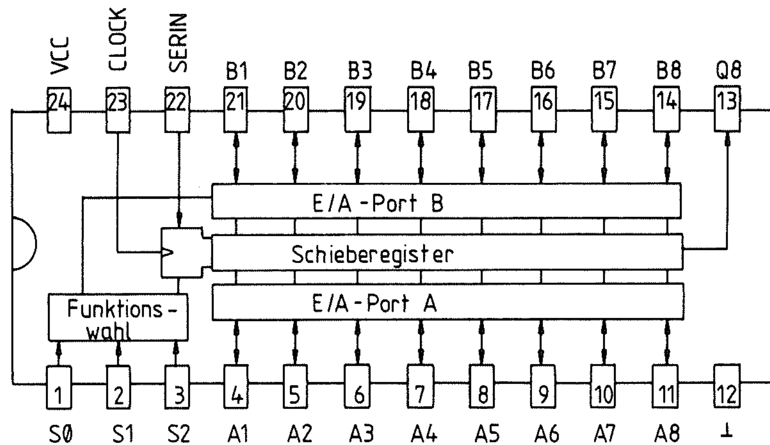
6.6 Nieuwe symbolen digitale logika

Toepassing van de nieuwe symbolen op een 8-bits universeel schuifregister type 74877, met de uitwerking in oude symbolen, alsook aansluitgegevens



Figuur 3/6.6-18: Het symbool

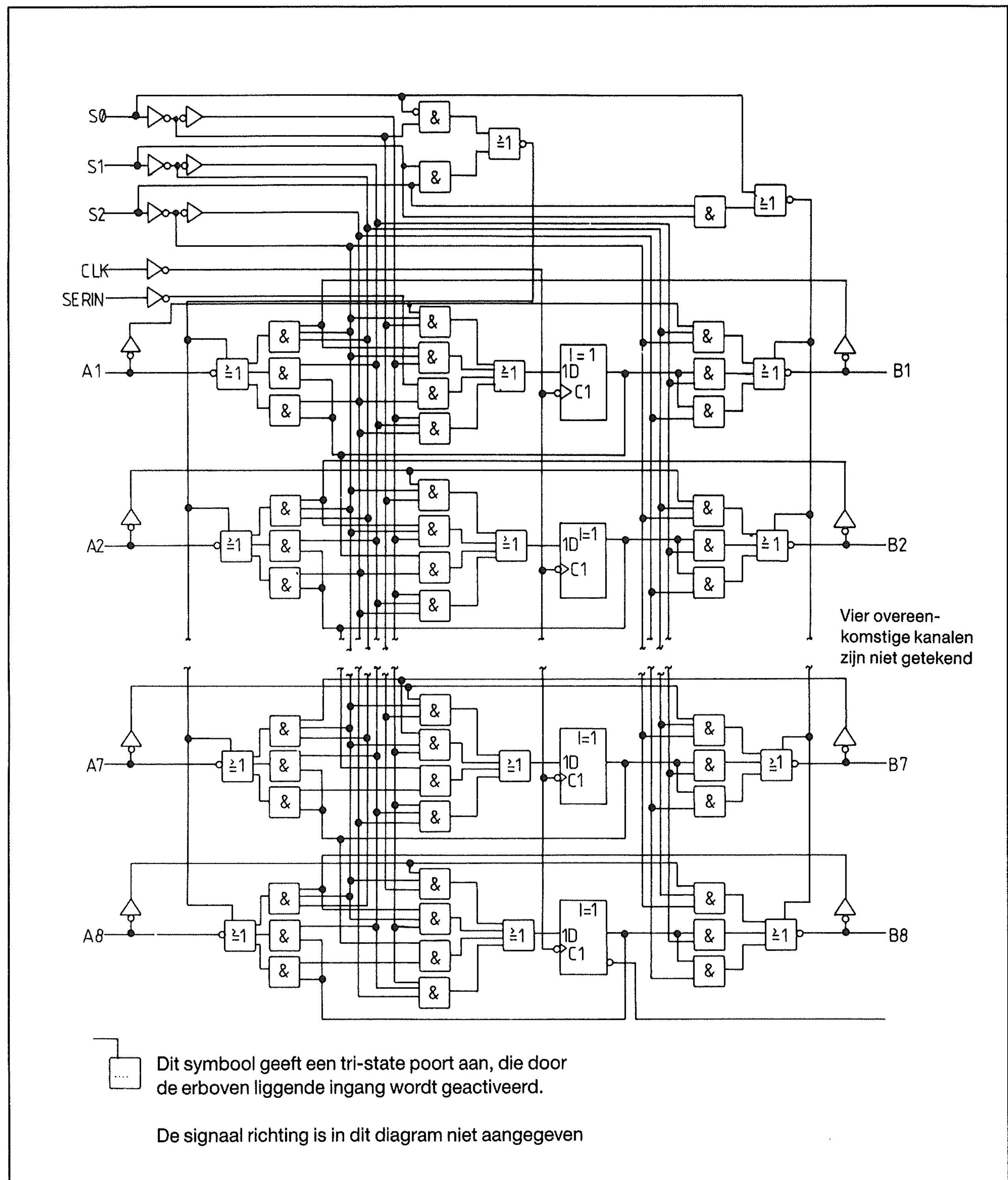
6.6 Nieuwe symbolen digitale logika



Toepassing:
Universele zender / ontvanger, signature analysis

Figuur 3/6.6-18b: Aansluitschema 74877

6.6 Nieuwe symbolen digitale logika



Figuur 3/6.6-18c: Het uitgewerkte schema van een 74877

6.6 Nieuwe symbolen digitale logika

3/6.7

Bistabiele elementen (flip-flops)

Flip-flops zijn schakelingen, die door hun functie een geheugenwerking hebben. Zij zijn in staat een eenmaal opgeslagen logische toestand vast te houden, ook nadat het stuursignaal, dat deze toestand veroorzaakte, is verdwenen.

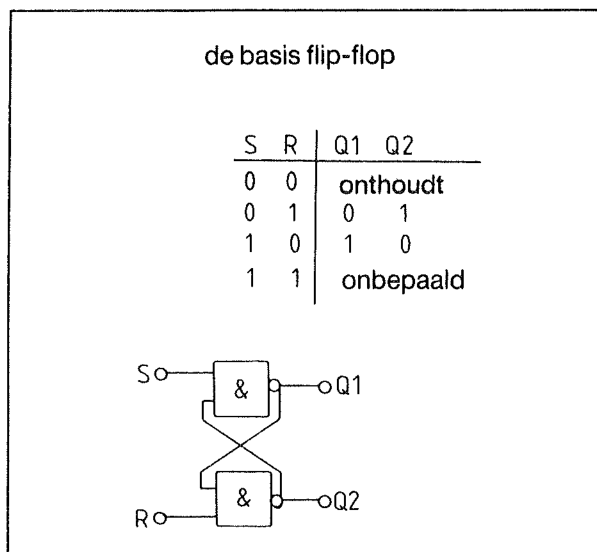
Het toepassingsgebied van flip-flops is in de huidige electronika niet meer weg te denken. Bekende toepassingen zijn: contactdenderonderdrukking, frequentiedelers, tellers, schuifregisters, tot en met grote zeer complexe geheugens. Door de eisen, die verschillende toepassingen aan flip-flops stellen, werden er in de loop der jaren vele verschillende soorten flip-flops ontwikkeld. Hoe een flip-flop data opslaat, welke typen er zijn en hoe die werken wordt in dit hoofdstuk uit de doeken gedaan. In de praktijk zijn tegenwoordig nagenoeg alle flip-flops in geïntegreerde vorm verkrijgbaar.

3/6.7.1 de basis flip-flop

Het basis principe van alle flip-flops, hoe gecompliceerd ook is in wezen gelijk. Deze eenvoudigste flip-flop versie noemen we hier de basis-flip-flop. De basis flip-flop kan uit NOR- of uit NAND-poorten zijn opgebouwd. In figuur 3/6.7.1-1 laat de NOR-poorten versie zien. De bedoeling is aan de uitgangen Q1 en Q2 een stabiele, doch tegengestelde logische toestand te scheppen.
Het is de terugkoppeling van de uitgangen, naar een ingang van de naastliggende poort, die voor de stabiele toestand zorgt.

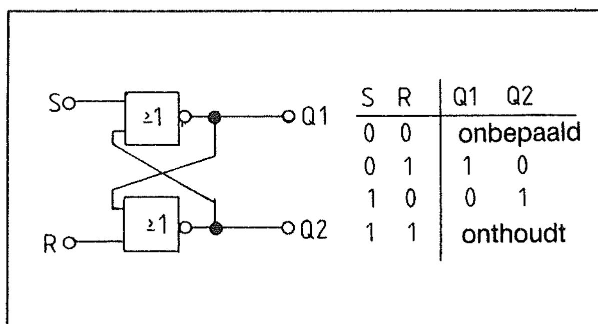
De ingang S (set) en R (reset) bepalen in welke stabiele toestand de flip-flop komt.
Als we de ingangen S en R op 0 houden, terwijl de voedingsspanning wordt aangezet, dan zal dat resulteren in een stabiele uitgangstoestand. Welke is echter niet te voorspellen. Met S = 1 en R = 0 "set" met de flip-flop, d.w.z. dat de Q1=1 en Q2=0 toestand ontstaat. Zelfs als men S weer 0 maakt, blijft deze toestand bestaan.
De flip-flop "onthoudt" de situatie.

Een verandering zal pas optreden, als S=0 en R=1. De flip-flop klapt dan om, d.w.z. de tegengestelde uitgangstoestand verschijnt, waardoor Q1=0 en Q2=1. De flip-flop is "ge-reset".
Ook deze toestand blijft stabiel nadat de oorzaak (R=1) weer weg genomen is (R=0).
Als beide ingangen gelijktijdig 1 zijn, dan ontstaat er een vreemde situatie.



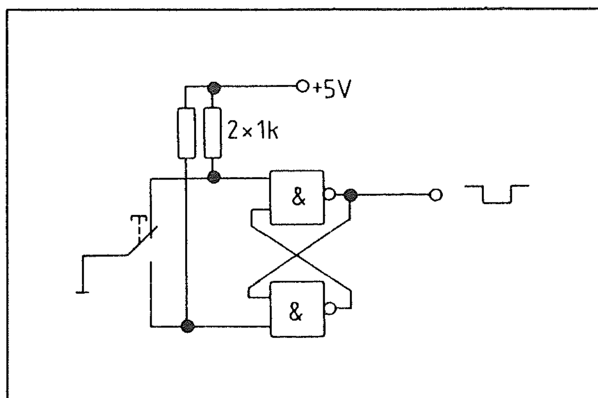
S	R	Q1	Q2
0	0	onthoudt	
0	1	0	1
1	0	1	0
1	1	onbepaald	

Figuur 3/6.7-1: Basis flip-flop met NOR poorten



S	R	Q1	Q2
0	0	onbepaald	
0	1	1	0
1	0	0	1
1	1	onthoudt	

Figuur 3/6.7-2: Basis flip-flop met NAND poorten



Figuur 3/6.7-3: Contact-dender-onderdrukking

Beide uitgangen willen hetzelfde logische niveau voeren, hetgeen door de terugkoppeling in tegenspraak is met de NOR functie van de individuele poorten. Deze situatie dient men dan ook te vermijden. Dikwijls wordt deze situatie aangeduid als een "verboden" toestand. Een stabiele uitgangstoestand ontstaat pas weer, als één der ingangen weer 0 wordt.

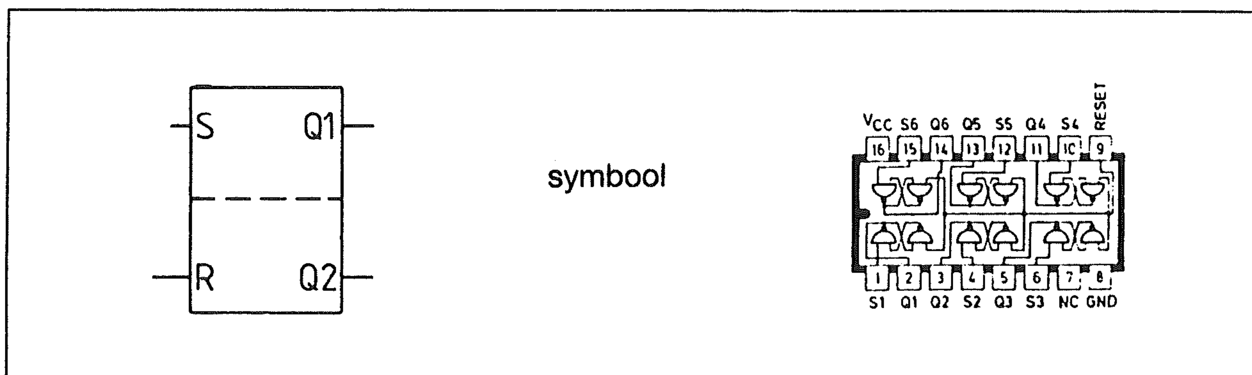
Als we ter vergelijking eens naar de NAND-uitvoering van de flip-flop kijken (figuur 3/6.7-2) dan zien we maar weinig verschil. De flip-flop wordt in dit geval ge-set en ge-reset met een logisch 0 niveau, de sturing is dus tegengesteld aan de sturing van de NOR-versie. De onbepaalde situatie treedt logischer wijze op bij R=0 en S=0.

De basisflip-flop is een toestand gestuurde flip-flop. Immers de uitgangen veranderen alleen, als er iets aan de ingangstoestand verandert. Zij hebben echter een ingangstoestand die vermeden dient te worden. Het toepassingsgebied van toestand-gestuurde flip-flops beperkt zich tot eenvoudige schakelingen, zoals contactdenderonderdrukking, detecteren of onthouden van een korte puls e.d. Hierbij is men er tamelijk zeker van, dat de "verboden" toestand niet zal optreden. Zie ook figuur 3/6.7.1-3. Het IC type 74118 bevat zes RS-flop-flops. De reset is gemeenschappelijk. Zie figuur 3/6.7.1-4.

3/6.7.2 Flank getriggerde flip-flops

Flank getriggerde flip-flops worden in de datahandboeken meestal met de Engelse term edge-triggerd aangeduid. Hier zullen we ons echter zo veel moge-

6.7 Bistabiele elementen (flip-flops)



Figuur 3/6.7-4: Symbool van een RS flip-flop en aansluitschema 74118

lijk van de Nederlandse term bedienen, omdat die in het kader van de bespreking goed aangeeft wat er gebeurt.
Flank getriggerde flip-flops reageren niet meer op een logisch niveau van een stuursignaal, maar op een stijgende of dalende flank hiervan. Slechts gedurende deze korte tijd is verandering van de uitgangstoestand mogelijk.

De rest van de tijd bevindt de flip-flop zich in de "onthoudt" toestand. De storingsgevoeligheid is dan ook aanzienlijk kleiner. Flank triggering kan op verschillende manieren.

In figuur 3/6.7.2-1: propagation delay (=looptijdvertraging) zijn enkele poorten voor de eigenlijke stuurpoort aangebracht. De verschillende pulsen hangen af van de gebruikte poorten. De pulstijd is de optelsom van de propagation delay van de individuele voorgeschakelde poorten. Voor TTL-poorten ligt de waarde van de propagation delay tussen de 10-20 ns. Variaties zijn type en fabrikant afhankelijk.

Een tweede mogelijkheid is weergegeven in figuur 3/6.7.2-2, waar een differentieatienetwerkje in het stuurleiding is opgenomen. Omdat de condensator via een e-kurve wordt geladen cq. ontladen, neemt de flanksteilheid van de uitgangspuls af. In de symbolen is een klokflanksturing aangeduid met een driehoekje. Een kringetje voor het driehoekje geeft aan, dat de ingang reageert op een dalende flank. De diode achter het differentiatienetwerk zorgt ervoor, dat er geen destructieve negatieve spanningen op de ingang van het IC terecht komen.

3/6.7.3 Geklokte flip-flops

In de moderne elektronica is het vaak van belang, dat bepaalde zaken tegelijkertijd gebeuren. Meerdere schakelingen, waaronder flip-flops moeten op bepaalde nauwkeurige tijdstippen schakelen. Om dit te bereiken wordt gebruik gemaakt van een speciaal stuursignaal, dat aan alle onderdelen wordt aangeboden, zodat deze gelijktijdig kunnen reageren. Dit heet synchrone sturing, ook wel kloksturing. Om dit te bereiken heeft men speciale flip-flops ontworpen. Deze flip-flops kunnen slechts hun uitgang veranderen als er een bepaald kloksignaal wordt gestuurd. We onderscheiden vier groepen:

1. Enkelvoudige toestand gestuurde flip-flop
2. Dubbelvoudige toestand gestuurde flip-flop
3. Enkele flank gestuurde flip-flop
4. Dubbele flank gestuurde flip-flop

6.7 Bistabiele elementen (flip-flops)

opwekken van pulsflanken

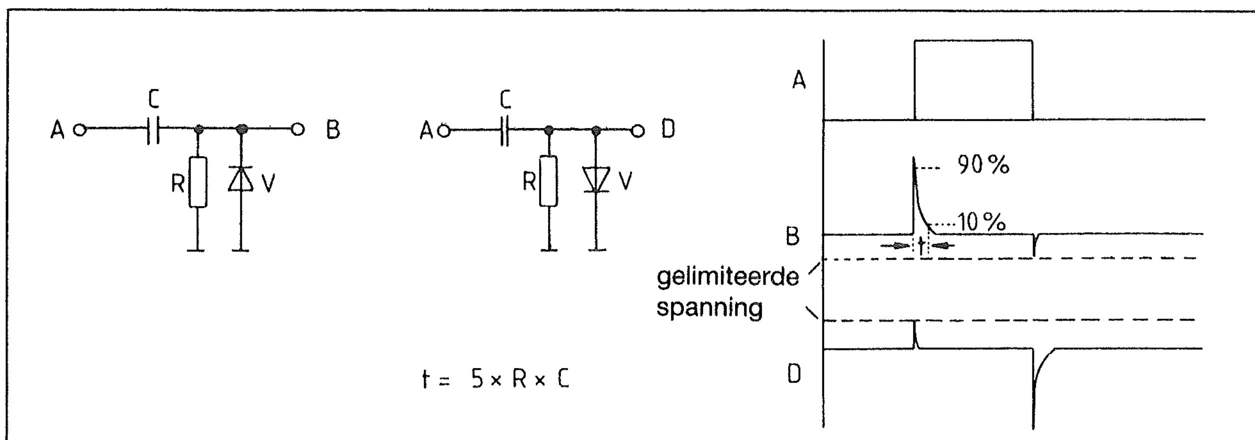
10ns 10ns 10ns

A F B C D E

t = 3 x poort propagation delay

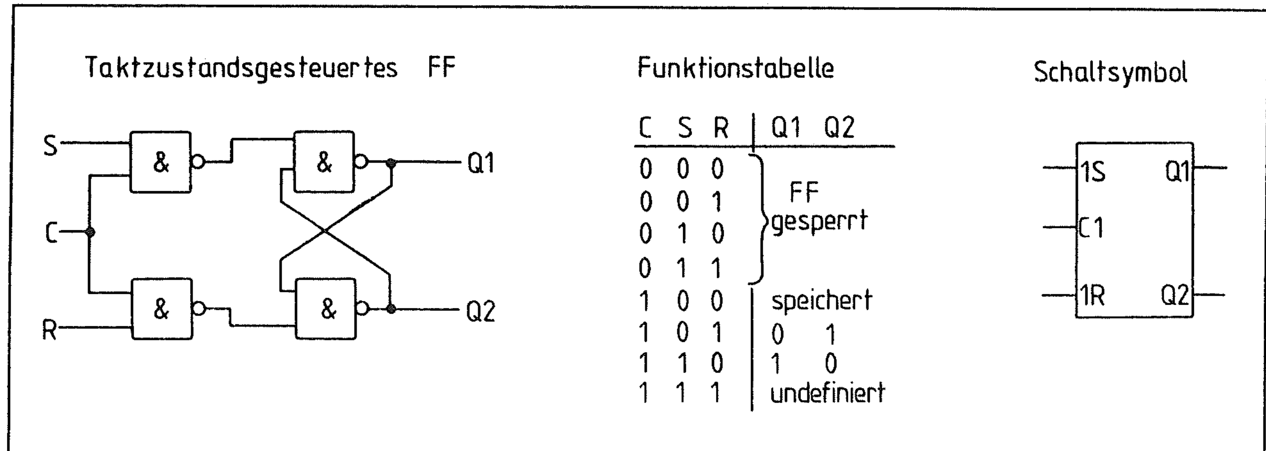
De propagation delay van de uitgangspoort blijft buiten beschouwing

Figuur 3/6.7.2-1: Pulsvertraging door propagation delay



Figuur 3/6.7.2-2: Pulsopwekking door differentiatie

6.7 Bistabiele elementen (flip-flops)



C	S	R	Q1 Q2
0	0	0	FF gesperrt
0	0	1	FF gesperrt
0	1	0	FF gesperrt
0	1	1	FF gesperrt
1	0	0	speichert
1	0	1	0 1
1	1	0	1 0
1	1	1	undefiniert

Figuur 3/6.7.3-1: Kloktoestand gestuurde flip-flop

Enkelvoudige toestand gestuurde flip-flop

Deze flip-flop is opgebouwd uit een basis flip-flop, met extra poorten aan de R- en S- ingangen.
Zie figuur 3/6.7.3-1: Kloktoestandgestuurde flip-flop. In dit geval is het klok signaal een enable (vrijgave) signaal. Deze flip-flop kent echter nog steeds een verboden toestand (in dit geval alle ingangen 1).

De D-flip-flop

Een D-flip-flop (D=delay=vertraging) is een uitgebreide versie van de basis flip-flop met een toestandgestuurde klok ingang. De set-ingang is via een inverter verbonden met de reset-ingang. Zonder klok is er voor deze flipflop geen mogelijkheid een toestand te onthouden. Duidelijk is te zien, dat deze flip-flop zijn uitgangen kan veranderen, zodra de voorgeschakelde NAND-poorten door de C-ingang zijn geactiveerd. Zie figuur 3/6.7.3-2. De D-ingang bereidt de uitgangstoestand voor. De C-ingang bepaald het moment van de omschakeling. In het symbool vinden we dit terug door de aanduiding van C als stuuringang met het nummer van de ingangen waarop dit stuursignaal betrekking heeft. Het hoeft dan ook geen betoog, dat we dit nummer terug vinden bij de D-ingang. In functietabellen is vaak de situatie voor en na de klokpuls weergegeven. Doorgaans wordt de situatie vlak voor de klokpuls aangegeven met t_n en de toestand na de klokpuls met t_{n+1}.

Een D-flip-flop wordt gebruikt voor de doorgave van ingangssignalen op een synchrone wijze.
Als we een D-flip-flop aansluiten als links in figuur 3/6.7.3-3, dan werkt de D-flip-flop als een synchrone 2-deler. De rechter figuur levert een aan/uit schakelaar met stabiele uitgangstoestanden.
Vaak worden D-flip-flops toegepast om de toestand te bufferen, d.w.z. de toestand die op de D-ingang aanwezig was ten tijde van de klok te bewaren. In het Engels worden ze ook wel "latch" flipflops genoemd, of kortweg "latches". Voorbeelden van D-flip-flops zijn bijv. een 7475.
Dit IC bevat vier van deze flip-flops. Een 7474 heeft 2 D-flip-flops met asynchrone Set- en Reset-ingangen die hier Preset en Clear worden genoemd. Asynchroon wil zeggen, dat deze ingangen onafhankelijk van de klok werken.

6.7 Bistabiele elementen (flip-flops)

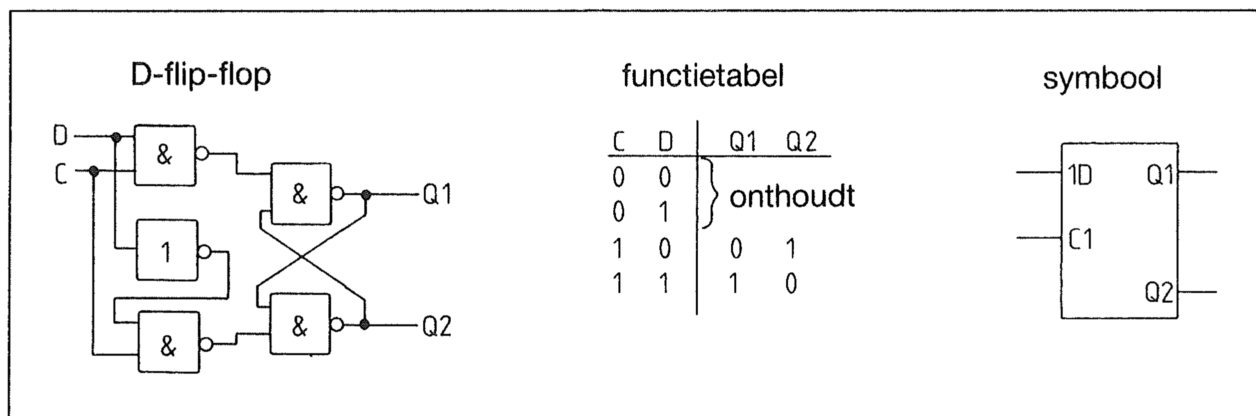
JK-flip-flops

De voortdurende vraag naar een flip-flop, waarvan de uitgangstoestand te allen tijde bepaald is, heeft geresulteerd in de ontwikkeling van JK-flip-flops. De JK-flip-flop kent evenals de D-flip-flop geen verboden toestand. De schakeling aan de ingang voorziet echter in een veel gevraagde mode. De schakeltechnische opbouw van een JK-flip-flop is te zien in figuur 3/6.7.3-4. De AND-poorten op de ingangen worden door de uitgangstoestand van Q1, resp. Q2 om en om vrij gegeven. De aanduiding van de ingangen met J en K is willekeurig en heeft geen verdere betekenis.

Als men aan beide ingangen J en K een logische 1 aanbiedt, dan schakelt de flip-flop bij elke klokpuls om. Dit noemt men "toggle". In de toggle-mode is de JK-flip-flop dus een gewone frequentiedeler.

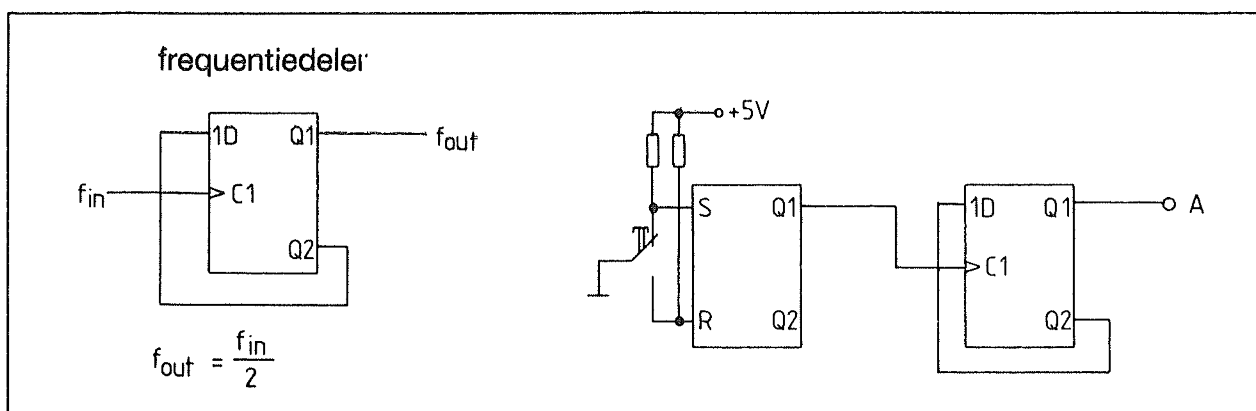
De functie komt in deze mode overeen met het voorbeeld van de D-flip-flop met terugkoppeling van de Q2-uitgang naar de D-ingang. Afhankelijk van de toepassing kan de JK-flip-flop als toestands- of als flank-gestuurde flip-flop zijn gemaakt.

Voorbeelden van JK-flip-flops kunt u vinden in deel 6 van dit boek.



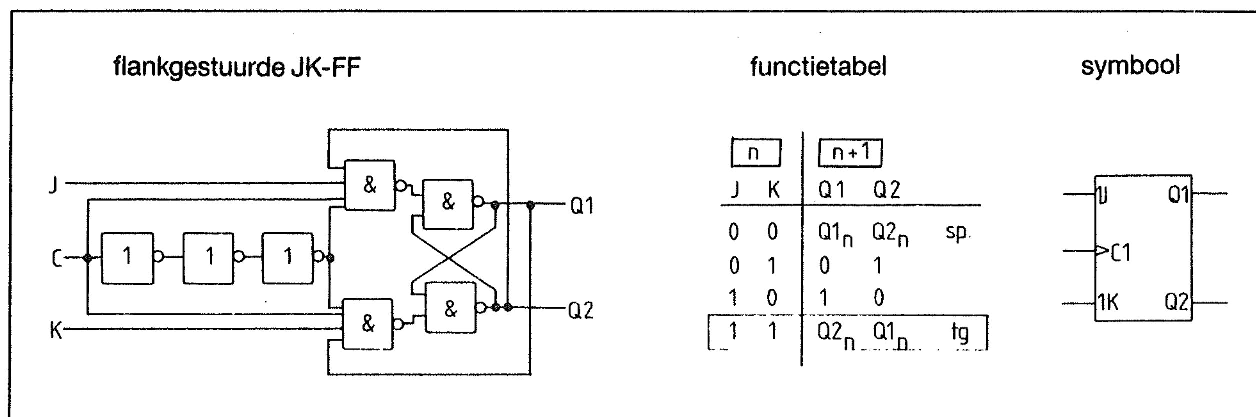
C	D	Q1	Q2
0	0	onthoudt	
0	1	onthoudt	
1	0	0	1
1	1	1	0

Figuur 3/6.7.3-2: De D(elay) flip-flop



Figuur 3/6.7.3-3: D-FF als deler en aan/uit schakelaar

6.7 Bistabiele elementen (flip-flops)



Figuur 3/6.7.3-4: Flankgestuurde JK-flip-flop

Het Master-Slave principe

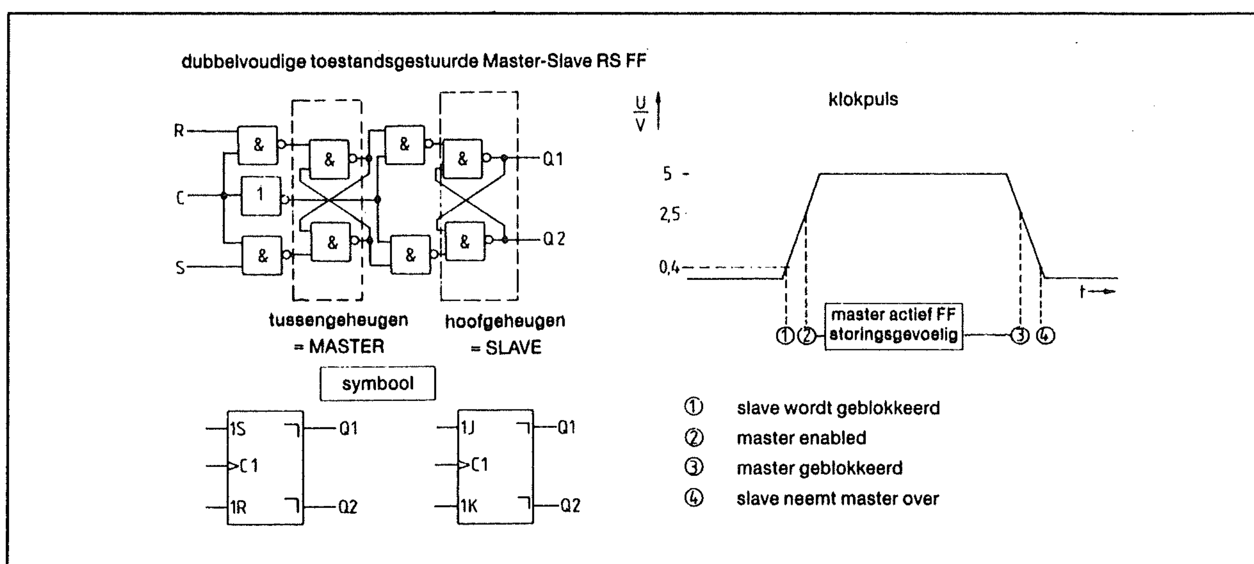
De aanduiding Master (heer) en Slave (slaaf) hebben niets te maken met rassendiscriminatie. Er wordt mee aangegeven, dat de flip-flop twee geheugenfuncties heeft.

Deze geheugens worden gelijkelijk, maar niet gelijktijdig geactiveerd. Zie voor de opbouw van een MS-flip-flop figuur 3/6.7.3-5: Het MS-principe. De werking wordt door de klok-ingang, of beter gezegd door de klokpuls bestuurd. Dit gebeurt in vier fasen:

Fase 1: Bij een stijgende flank op de klok word de slave (hoofdgeheugen) van de master (tussengeheugen) gescheiden.

Fase 2: De Master wordt vrijgegeven en neemt de ingangsinformatie over.

Fase 3: Bij de dalende flank op de klok neemt de slave de informatie



Figuur 3/6.7.3-5: Het Master-Slave principe

over en presenteert ze op zijn uitgangen.

De flip-flop in figuur 3/6.7.3-5 behoort tot de groep van dubbelvoudige toestandsgestuurde flip-flops, nadeel van deze flip-flops is de storingsgevoeligheid tussen de fasen 2 en 3. Gedurende die periode wordt de informatie overgenomen in de slave-flip-flop. Ook eventuele stoorsignalen.

Dit nadeel kan men vermijden als men de flip-flop flankgestuurd maakt (dubbel flank gestuurd), of de stuuringangen vlak na de overname van de data van de ingangen blokkeert (data-lock-out). Beide mogelijkheden worden in IC's toegepast.

Als de master een RS flip-flop is, dan zou een verboden toestand mogelijk zijn. Dat is de reden, dat de meeste master-slave flip-flops van het JK-type zijn. Voor de slave kan wel een RS flip-flop worden gebruikt. De slave is van buitenaf toch niet aan te sturen en de interne schakeling maakt het optreden van de verboden toestand onmogelijk. Asynchrone set- resp. reset-ingangen zijn altijd verbonden met de slave. Zij werken onafhankelijk van de klok.

Uit het tijddiagram van figuur 3/6.7.3-6 komt naar voren, dat de slave het kloksignaal een halve klokperiode vertraagd. Dit gegeven opent een groot aantal toepassingen voor master-slave flip-flops.

Door gebruik van de klok onafhankelijke R- en S-ingangen kan de schakeling op elk willekeurig moment in een bepaalde uitgangstoestand worden gebracht.

3/6.7.4 Het overzicht

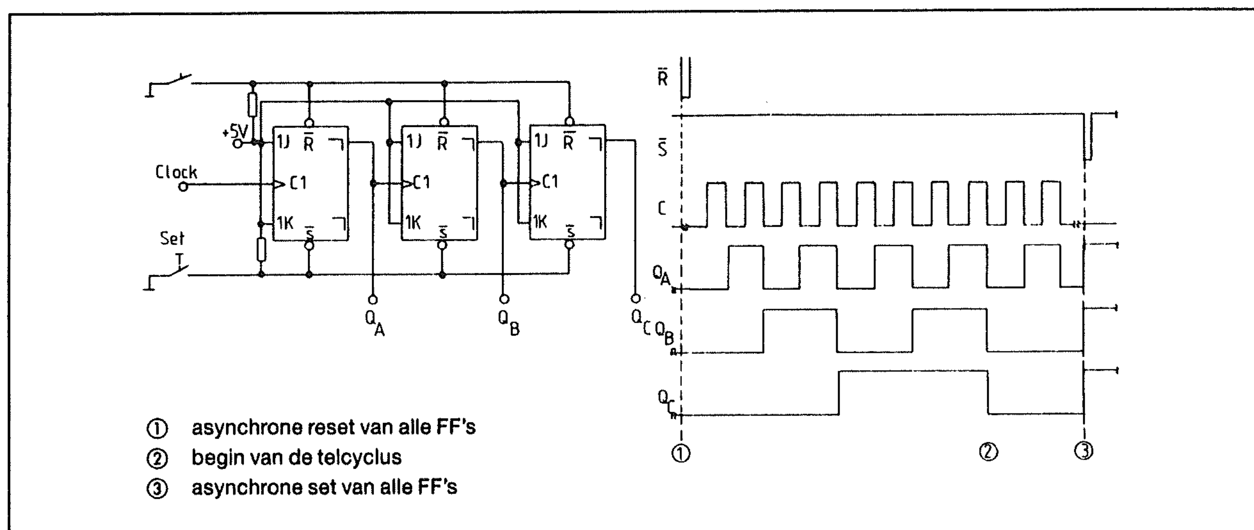
Zoals we zien is er een groot aantal verschillende flip-flops. Om het overzicht te behouden hebben we ze in een tabel bijeengebracht, gerangschikt naar functie en in groepen ingedeeld. De indeling is gemaakt aan de hand van het soort sturing. Zie tabel 3/6.7.4-1.

Bij het ontwerpen van schakelingen is het van belang hoe de individuele flip-flops op de stuursignalen reageren en aan welke ingangscondities moet worden voldaan om een bepaalde functie te verwezenlijken. Dit is terug te vinden in tabel 3/6.7.4-2.

Als u met veel nadenken en moeite uiteindelijk heeft vastgesteld wat voor soort flip-flop u nodig hebt, dan blijkt deze in datahandboeken niet te vinden. Wanhoop dan niet, maar bestudeer tabel 3/6.7.4-3: Aanpassen van flip-flops. Deze tabel laat zien hoe men van een type flip-flop de functie zo kan veranderen, dat het een ander type flip-flop wordt.
De type aanduiding van flip-flops is afhankelijk van het land van herkomst en het jaar van uitgave van een datahandboek. Vooral de wijze waarop de uitgangsfunctie in de functietabellen is aangegeven varieert.

Bij het inschakelen van de voedingsspanning is altijd het zwart opmlijnde veld logisch 1; d.w.z. dat de flip-flop bij het inschakelen is gereset. (De zogenaamde voorkeursstand). Schakeltechnisch wordt dit bereikt, door bij de fabrikage van het IC de uitgangstransistoren verschillend te doteren, zodat een bepaalde stand bij het inschakelen wordt afgedwongen. In de nieuwe

6.7 Bistabiele elementen (flip-flops)

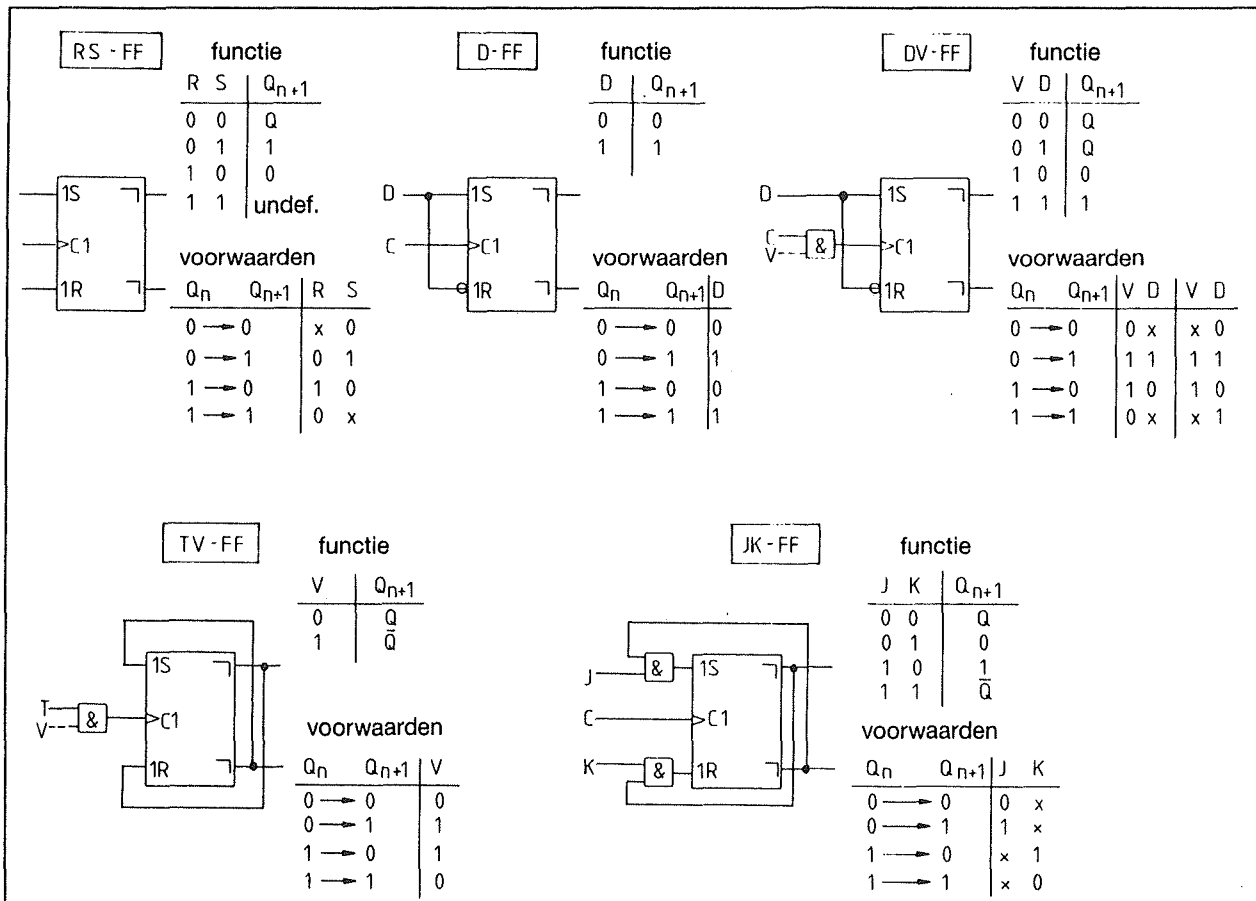


Figuur 3/6.7.3-6: Asynchrone reset van alle FF's

klok / input soort	niet geklokt		geklokt			
	toestands-sturing	flank-sturing	enkele toe-standssturing	dubbele toe-standssturing	enkele flanksturing	dubbele flanksturing
zonder inputs				T	>T	>T
RS flip-flop	S R	>S >R	1S C1 1R	1S C1 1R	1S >C1 1R	1S >C1 1R
JK flip-flop		>J >K		1J C1 1K	1J >C1 1K	1J >C1 1K
D flip-flop			1D C1	1D C1	1D >C1	1D >C1

Tabel 3/6.7.4-1: Overzicht van de belangrijkste soorten flip-flops

6.7 Bistabiele elementen (flip-flops)



Figuur 3/6.7.4-2: Functie tabellen van FF's

norm voor symbolen (IEC std 91-1984) worden zelfs de verboden toestanden deels gedefinieerd. Ook dit heeft te maken met voorkeurstoestanden.

Men doet er goed aan bij het ontwerpen altijd te zorgen, dat alle ingangen op een gedenieerd niveau zijn aangesloten. Als de flip-flops klok-onafhankelijke set- of reset-ingangen hebben dan moeten deze ingangen worden voorzien van externe pull-up of pull-down weerstanden naar de voeding cq. aarde. Verzaakt men dit, dan kunnen er onverwachte problemen ontstaan, die ook nog eens zeer moeilijk zijn te traceren.

Probeer met AND-poorten een gelijktijdig activeren van de set- en reset-ingangen te vermijden.

Flip-flops uit de TTL-reeks hebben een propagation delay van 30-120ns. De maximale klokfrequenties liggen tussen de 8 en 20 MHz. De snelste logika is ECL. Deze logika heeft echter twee voedingen nodig en het is niet eenvoudig de niveaus aan TTL-niveaus aan te passen.

Bij C-MOS flip-flops is de propagation delay afhankelijk van de voedingsspanning. Bij 100 omschakelingen per se-

6.7 Bistabiele elementen (flip-flops)

	J K	R S	D	T	binaire delers	karakteristieke vergelijking
J K		$J = S \quad K = R$	$J = D \quad K = \overline{D}$	$J = K = T$	$J = K = 1$	$Q = (J \wedge \overline{Q}_n) \vee (\overline{K} \wedge Q_n)$
R S	$S = J \wedge \overline{Q} \quad R = K \wedge Q$		$S = D \quad R = \overline{D}$	$S = T \wedge \overline{Q} \quad R = T \wedge Q$	$S = \overline{Q} \quad R = Q$	$Q = S \vee (\overline{R} \wedge Q_n)$ wobei $S \wedge R = 0$
D						$Q = D$
T						$Q = (T \wedge \overline{Q}_n) \vee (\overline{T} \wedge Q_n)$

$Q = Q_{n+1}$

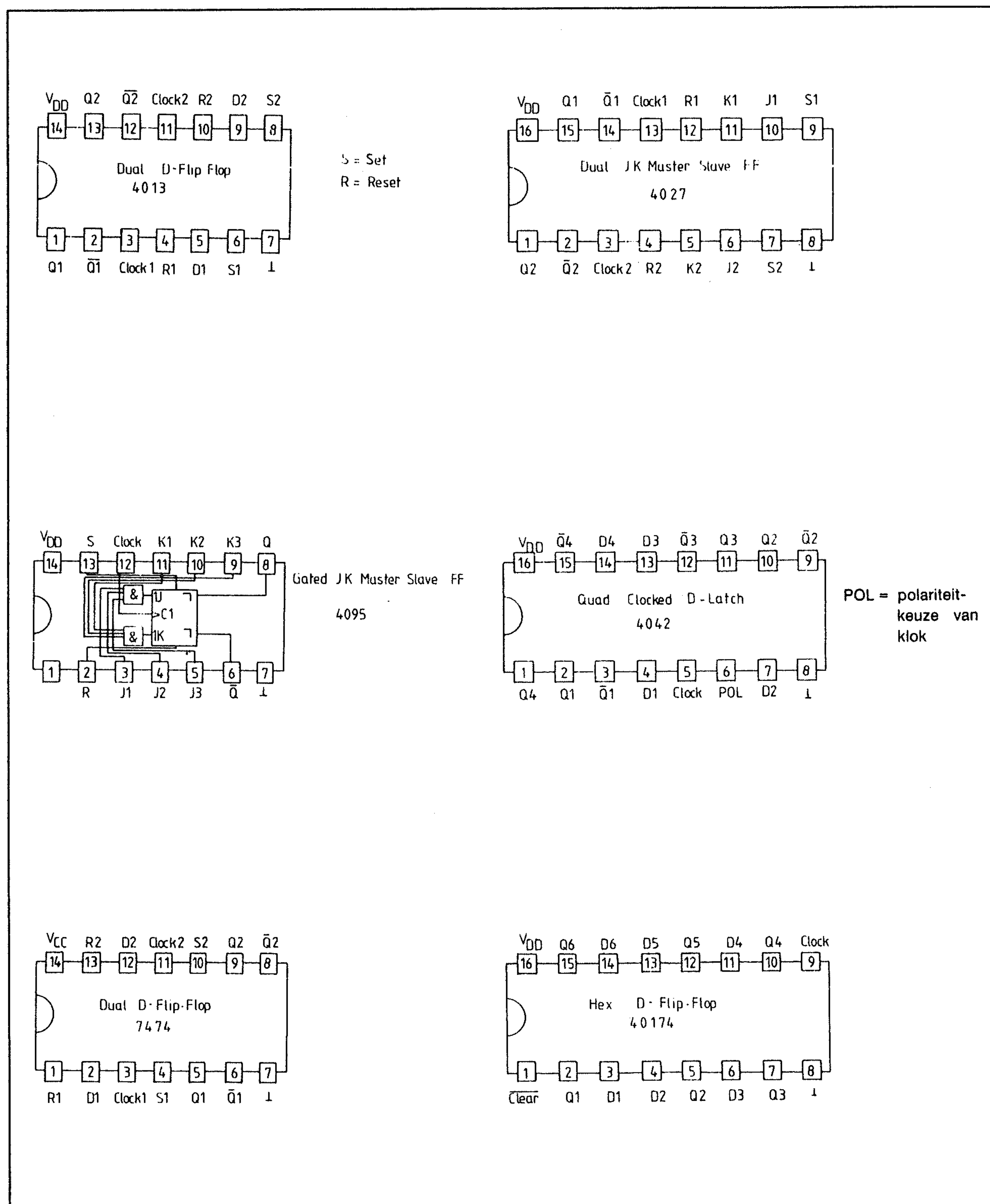
Figuur 3/6.7.4-3: Aanpassen van flip-flops

conde is het opgenomen vermogen al aanzienlijk groter dan in de rusttoestand. Bij een miljoen omschakelingen per seconde is het opgenomen vermogen zelfs groter, dan het opgenomen vermogen van LS-TTL IC's.

Voor langzame toepassingen is C-MOS met een 5V voeding de beste keuze. Voor hogere snelheden kan men de voeding verhogen naar 10V of 12V. De propagation delay neemt toe met de temperatuur en is ook afhankelijk van de capacitieve belasting. Langzaam stijgende of dalende signalen kunnen oscillaties of dubbeltriggering veroorzaken. Een slecht geregelde of ontkoppelde voeding kan dit probleem nog versterken, daar dan het ingangssignaal ook nog eens met de voeding mee varieert.

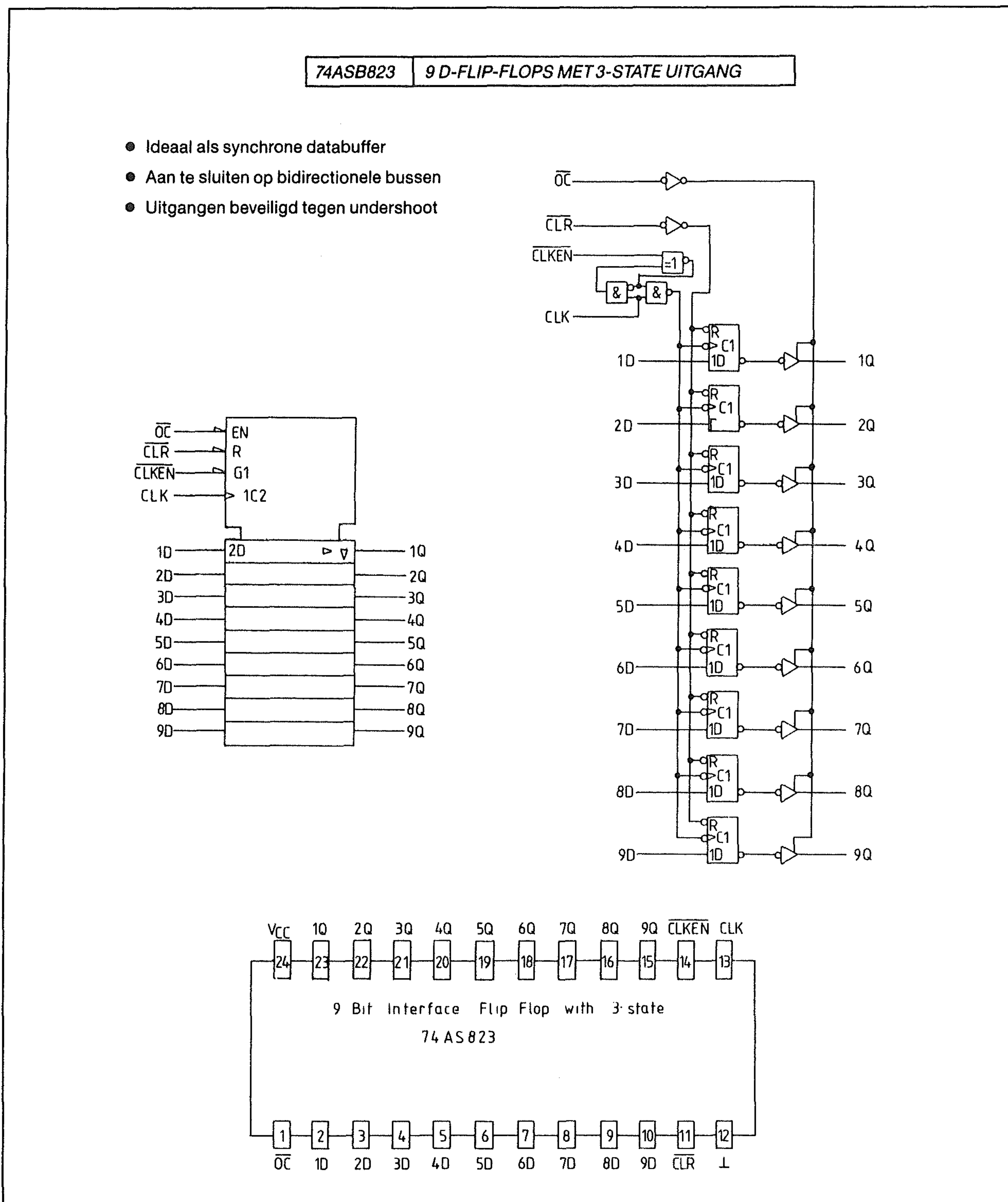
Veel halfgeleider fabrikanten leveren IC's in de 40-serie. De variatie in snelheid en fan-out is tamelijk groot. Als u dus een bepaald type niet snel genoeg vindt, kijk dan eens in het datahandboek van een andere fabrikant.

6.7 Bistabiele elementen (flip-flops)



Figuur 3/6.7.4-1: Een selectie geïntegreerde FF's met aansluitschema

6.7 Bistabiele elementen (flip-flops)



Figuur 3/6.7.4-2: Dit IC kan worden gebruikt in microprocessor systemen, als uitleesbuffer van externe bussen. Aansturing kan via de adresdecoder gebeuren

3/6.8

Digitale codes en omzetters

Als men met digitale systemen werkt, dan moet men een manier bedenken om gegevens in een bepaalde vorm in te geven. Ook zijn de codes die in de digitale techniek worden toegepast niet zomaar te lezen.

De gegevens moeten dus gecodeerd en gedecodeerd worden. Laten we het proces van deze codering en het omzetten van de ene code in de andere eens aan een nadere beschouwing onderwerpen.

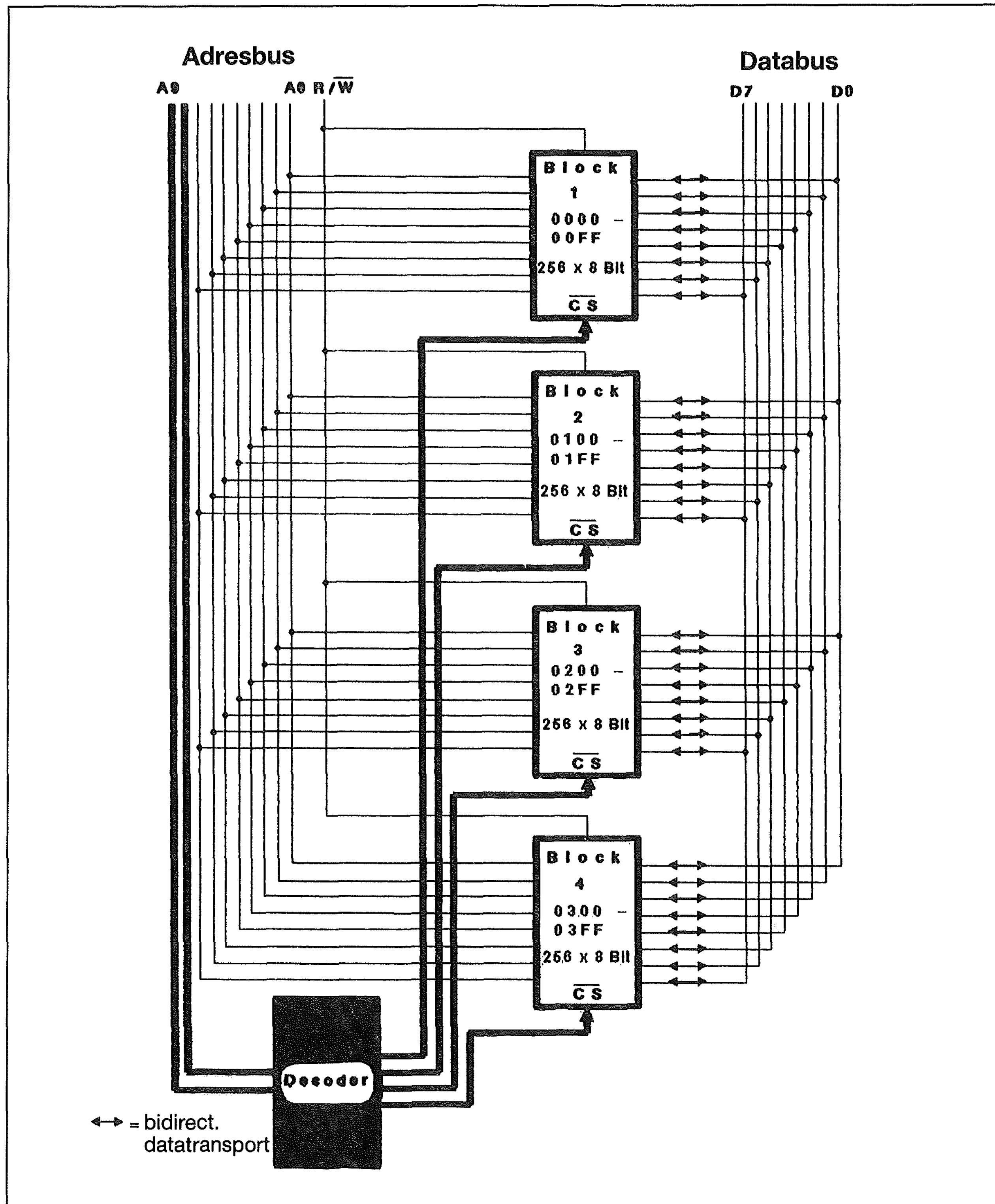
In principe is een ongelimiteerd aantal codes mogelijk. Welke code men kiest is dan afhankelijk van de toepassing waarvoor men de code wenst te gebruiken. Gelukkig is er echter door de praktijk een zekere limitatie opgetreden, zodat het aantal codes toch nog valt te overzien. Er zijn verschillende codes in omloop voor bijvoorbeeld reken, displays, zevensegment displays etc. De vertalers van de ene naar de andere code worden omzetters genoemd. In "goed" nederlands worden echter ook vaak de engels termen "decoder" en "encoder" gebruikt. Een code-omzetter zet een bepaalde code die op de ingang wordt aangeboden om in een andere code op de uitgang.

Vaak vinden we code-omzetters ingebouwd in display-drivers. In deze schakelingen worden ze echter niet als zodanig erkend of genoemd.

Ook worden code-omzetters gebruikt in geheugenschakelingen. In deze schakelingen worden ze vaak gebruikt om bepaalde geheugencellen of blokken te selecteren en/of vrij te geven via chip-enable of chip-select ingangen. Op de ingang wordt doorgaans adresinformatie aangeboden. In figuur 3/6.8-1 wordt aanschouwelijk gemaakt, hoe een dergelijke adresselectie in zijn werk gaat.

Wat is een code?
Het woord code is afkomstig uit het berichtenverkeer. Een opeenvolging van signalen wordt daarbij middels een codetabel weer omgezet in leesbare tekst. De tabel bevat de afspraken over de betekenis van de signalen. Deze betekenis is de code. Meestal is de betekenis van een serie signalen steeds gelijk. Een uitzondering is encryptie, waar de betekenis telkens anders kan zijn. Het zal duidelijk zijn, dat om codes te ontcijferen de ontvangende partij op de hoogte moet zijn van de regels die in de code zijn vastgelegd. Een welbekend voorbeeld van codes is het Morse-alfabet.
In de digitale techniek is er een aantal codes in gebruik, elk met zijn eigen regels. Meestal is de gecodeerde boodschap onleesbaar. Pas na omzetting wordt de boodschap duidelijk.

6.8 Digitale codes en omzetters



Figuur 3/6.8-1: Selectie van geheugenblokken met behulp van decoders.

6.8 Digitale codes en omzetters

Met de toeneming van de complexiteit van de digitale techniek en de integratietechnieken, werd de mogelijkheid geschapen om steeds ingewikkelder coderingen toe te passen.

Coderingsvormen

1. De gesloten codering

Hierbij wordt aan elk decimaal getal een unieke binaire code toegekend. De code is steeds afhankelijk van de totale waarde van het decimale getal en niet van het aantal cijfers in het getal. Een voorbeeld vindt u in figuur 3/6.8-2: Codering van het decimale getal 55 in binaire code.

Deze vorm zou u nu reeds bekend moeten voorkomen, daar we hem al eerder in de getallen stelsels hebben behandeld.

2. De open (plaats bepaalde) codering

Wat ligt er meer voor de hand, dan elk cijfer van een getal te coderen. Het stelsel waarin wij rekenen (decimaal) is immers ook op deze wijze gecodeerd.

decimaal getal			binaire code plaatsafhankelijke waarde					
10	1		32	16	8	4	2	1
5	5 $_{(10)}$	=	1	1	0	1	1	1

Figuur 3/6.8-2: De gesloten codering.

Hierop doorbordurend komt men uit op de codering, zoals die is voorgesteld in figuur 3/6.8-3. Hierin ziet u weer de omzetting van het decimale getal 55 in de 8421 code, ook wel de BCD-code genoemd. (binary coded decimal). Elk cijfer (digit) van het decimale getal wordt omgezet in een cijfer (digit) van de code.

Deze codering heeft als voordeel, dat zij makkelijker te lezen is, omdat de structuur van het decimale getal blijft behouden. Het voordeel van de eerder besproken binaire code is echter, dat er minder plaatsen (bits) nodig zijn om de waarde aan te geven, en dat de regels om met deze getallen te kunnen rekenen eenvoudiger zijn.

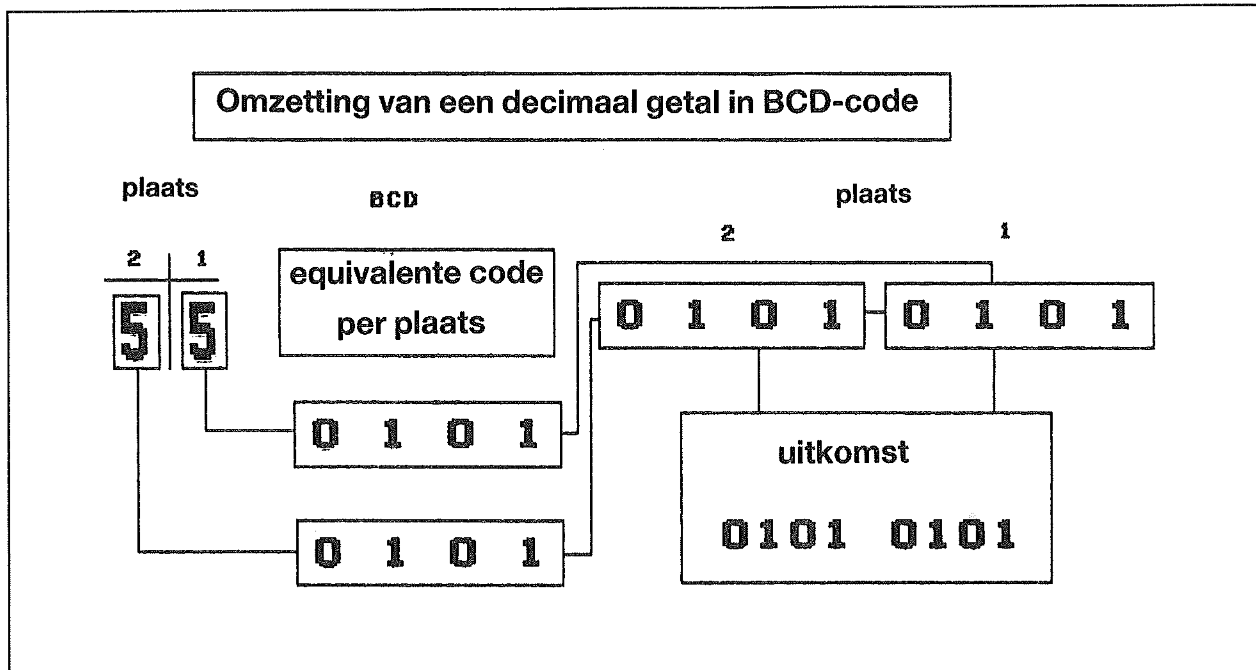
3. De minimaal code

Een verdere onderverdeling in codes is te vinden in het benodigde aantal plaatsen (digits). Bij de minimaal code is het aantal plaatsen onafhankelijk van de grootte van het te coderen getal. Alleen het maximaal te coderen getal ligt vast, door de bepaling van hoeveel plaatsen men wenst te gebruiken. De BCD-code is een bekend voorbeeld van een minimaal code.

4. De optimaal code

Bij optimaal codes varieert het aantal plaatsen (digits). De meest gebruikte tekens krijgen een korte code en weinig gebruikte tekens krijgen een langere code. De Morse code is een goed voorbeeld hiervan. Optimaal codes worden gebruikt, als er berichten moeten worden overgezonden, waarin bepaalde tekens (codes) met een veel grotere frequentie voorkomen, dan andere.
Dit is doorgaans het geval met tekst. Het zal duidelijk zijn, dat hiermee een tijdsbesparing kan worden gerealiseerd. Een optimaal code kent echter geen of tamelijk complexe regels en werd in het verleden in de digitaal techniek ook nauwelijks toegepast. Omdat tekstverwerking en teksttransport in de moderne computers een

6.8 Digitale codes en omzetters



Figuur 3/6.8-3: De plaatsafhankelijke codering.

veel voorkomend verschijnsel is en grote hoeveelheden tekst ook moeten worden opgeslagen, heeft men algoritmes ontwikkeld om, gebruikmakend van optimaal codes, de benodigde opslagruimte te beperken en de transport tijden korter te maken. Bij tekst zijn 4 op 1 reducties geen uitzondering.

De in de digitale technieken gebruikelijke codes.

De 8421-code

De 8421-code is een additieve code, d.w.z., dat elke plaats (bit) binnen de code een bepaalde waarde is toegekend. De decmimale waarde van een gecodeerd getal vind men door de plaats-waarden van elk bit dat logisch een is, bij elkaar op te tellen. De 8421-code komt overeen met de binaire code. De plaatswaarden van de bits lopen van rechts naar links en kunnen naar believen worden uitgebreid. De waarde volgt uit de wiskundige vergelijking: plaatswaarde = 2^n waarbij n de plaats van het bit is. NB. het meest rechtse bit is aangeduid met plaats 0. Zie tabel 3/6.8-1.

De BCD-code.

Deze onderscheidt zich op slechts twee punten van de 8421-code. Ten eerste heeft de BCD-code altijd 4 plaatsen per cijfer en ten tweede zijn alleen de getallen 0 t/m 9 (decimaal) gedefineerd.

Het nadeel van deze code is het grote aantal benodigde digits. Omdat er 6 niet gedefinieerde waarden zijn, is rekenen in deze code tamelijk complex. We zullen hier later nog wel eens op terug komen.

6.8 Digitale codes en omzetters

BCD-codes worden voornamelijk gebruikt bij displays, omdat de structuur van de getallen direct herkenbaar is.

De Aiken-code.

Ook dit is een additieve code, waarbij de redundante (niet gedefinieerde) waarden echter in het midden van de codetabel liggen. Deze redundante waarden kunnen wel voorkomen, maar resulteren doorgaans in foutieve aanduidingen.

De Exxes-3 code

Bij deze code kan men de waarde niet aflezen van de plaats van de bits die log. 1 zijn. Het is een combinatiecode. Je zou ook kunnen zeggen, dat het een afspraakcode is voor cijfers.

Alleen de waarden 0 tot 9 decimaal kunnen worden weergegeven. De overige zes combinaties zijn redundant en niet gedefinieerd. Bij deze code liggen ze in het bovenste en onderste bereik.

De Gray-kode.

Dit is een bijzondere code. Het is de enige code waarbij bij de overgang naar de volgende decimale waarde slechts een bit van logische waarde verandert. Deze code wordt ook wel ripple-code of binary ripple-code genoemd.

Bij de codering van decimale getallen naar Gray-code, moet eerst het decimale getal in de binaire waarde worden omgezet.
Het binaire getal en het Gray-code getal moeten hetzelfde aantal plaatsen (digits) hebben.

Bij de omzetting van decimaal naar Gray-code worden naast elkaar staande bits van links naar rechts bekeken. Bij decodering van Gray-code naar decimaal juist andersom. Zie tabel 3/6.8-2.

De Gray-code wordt veel toegepast in analoog-naar-digitaal-converters en is ook zeer geschikt om transmissiefouten te analyseren.

6.8 Digitale codes en omzetters

Digitale codes

dec. getal	binaire code –	BCD-code 8-4-2-1 code	dec. getal	Aiken code	dec. getal	Exxess-3-code
plaats-waarde	8 4 2 1	8 4 2 1		2 4 2 1		geen
0	0 0 0 0	0 0 0 0	0	0 0 0 0		PT
1	0 0 0 1	0 0 0 1	1	0 0 0 1		PT
2	0 0 1 0	0 0 1 0	2	0 0 1 0		PT
3	0 0 1 1	0 0 1 1	3	0 0 1 1	0	0 0 1 1
4	0 1 0 0	0 1 0 0	4	0 1 0 0	1	0 1 0 0
5	0 1 0 1	0 1 0 1		PT	2	0 1 0 1
6	0 1 1 0	0 1 1 0		PT	3	0 1 1 0
7	0 1 1 1	0 1 1 1		PT	4	0 1 1 1
8	1 0 0 0	1 0 0 0		PT	5	1 0 0 0
9	1 0 0 1	1 0 0 1		PT	6	1 0 0 1
10	1 0 1 0	PT		PT	7	1 0 1 0
11	1 0 1 1	PT	5	1 0 1 1	8	1 0 1 1
12	1 1 0 0	PT	6	1 1 0 0	9	1 1 0 0
13	1 1 0 1	PT	7	1 1 0 1		PT
14	1 1 1 0	PT	8	1 1 1 0		PT
15	1 1 1 1	PT	9	1 1 1 1		PT

Tabel 3/6.8-1: De verschillende codes.

6.8 Digitale codes en omzetters

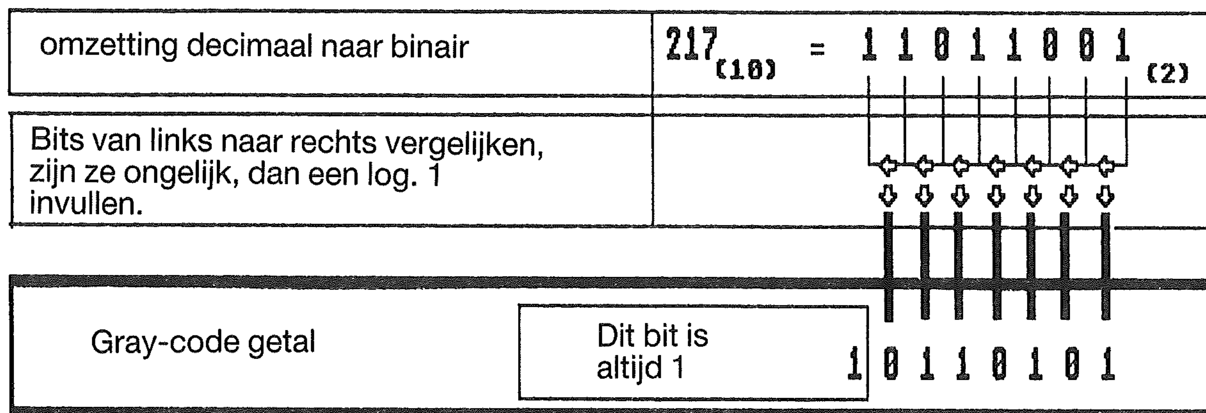
	BCD-code 8-4-2-1-code	Aiken code	Exxes-3-code
Omzetting	0-9 komt overeen met binaire code	0-4 komt overeen met binaire code 5-9 komt overeen met binaire code + 6	0-9 komt overeen met binaire code + 3
Voordeel	eenvoudige omzetting overzichtelijk	overzichtelijk en te complementeren	overzichtelijk, te complementeren, geen constante 0 of 1
Nadeel	constant 0, niet te complementeren	constante 0 en 1	

Opmerking:

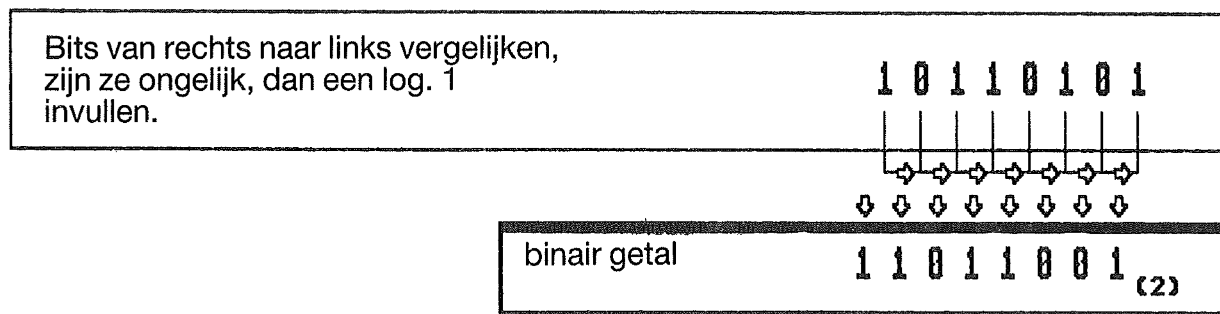
Coderingsmogelijkheden, zoals bovengenoemd, waarbij een aantal waarden niet zijn gedefinieerd, noemt men redundant. De niet gedefinieerde codes noemt men de relevantie. In bovengenoemde gevallen gaat het daarbij om zes pseudo-decimale getallen (PD). Een PD is een technisch mogelijke, maar niet gedefinieerde code.
Het resultaat kan van alles zijn.

6.8 Digitale codes en omzetters

Werking van de Gray-code



Gray-code naar binair



Tabel 3/6.8-2: Codering en decodering van Gray-code.

6.8 Digitale codes en omzetters

decimaal	binair					Gray-code				
getal	E	D	C	B	A	E	D	C	B	A
0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	1	0	0	0	0	1
2	0	0	0	1	0	0	0	0	1	1
3	0	0	0	1	1	0	0	0	1	0
4	0	0	1	0	0	0	0	1	1	0
5	0	0	1	0	1	0	0	1	1	1
6	0	0	1	1	0	0	0	1	0	1
7	0	0	1	1	1	0	0	1	0	0
8	0	1	0	0	0	0	1	1	0	0
9	0	1	0	0	1	0	1	1	0	1
10	0	1	0	1	0	0	1	1	1	1
11	0	1	0	1	1	0	1	1	1	0
12	0	1	1	0	0	0	1	0	1	0
13	0	1	1	0	1	0	1	0	1	1
14	0	1	1	1	0	0	1	0	0	1
15	0	1	1	1	1	0	1	0	0	0
16	1	0	0	0	0	1	1	0	0	0
17	1	0	0	0	1	1	1	0	0	1
18	1	0	0	1	0	1	1	0	1	1
19	1	0	0	1	1	1	1	0	1	0
20	1	0	1	0	0	1	1	1	1	0

Tabel 3/6.8-3: Functie tabel van de Gray-code.

Foutcorrectie codes

Datatransmissie is een der belangrijkste toepassingsgebieden voor codering. Juist bij communicatie van systeem naar systeem is het van belang, dat eventueel optredende fouten worden herkend en dat er een waarschuwing volgt. Bij ernstige niet te herstellen verminkingen kan het nodig zijn, dat het zendende systeem de data nogmaals zendt.

Men kan fouten detecteren, door alle overgezonden codes bij elkaar op te tellen, zonder carry, de som ook over te zenden en aan de andere zijde dezelfde bewerking toe te passen. De som moet in beide gevallen hetzelfde resultaat opleveren. Deze som wordt vaak aangeduid met de Engelse term "checksum". Ook is het mogelijk, bij elk getal een extra bit mee te zenden. De zender bepaalt de waarde van dit bit aan de hand van het aantal log. enen, dat in de echte code voorkomt. Dit extra bit wordt het pariteitsbit genoemd. Bij "even" pariteit, zal de zender dit bit logisch 1 maken, als het aantal enen in de code oneven is, zodat het totaal aantal enen in de code inclusief het pariteits bit altijd even is. Bij "oneven" pariteit wordt er door de zender op overeenkomstige wijze voor gezorgd, dat het aantal enen oneven is. Vaak zult u de Engelse termen "even" resp. "odd" parity tegenkomen. Zie ook figuur 3/6.8-4.

In de praktijk komen tegenwoordig nog uitgebreidere foutcontrole-systemen voor. Bij sommige systemen is het zelfs mogelijk optredende fouten zonder hertransmissie te corrigeren. Bij moderne compact-disk apparaten kunnen zo tot 32 bits van de code worden getest en gecorrigeerd.

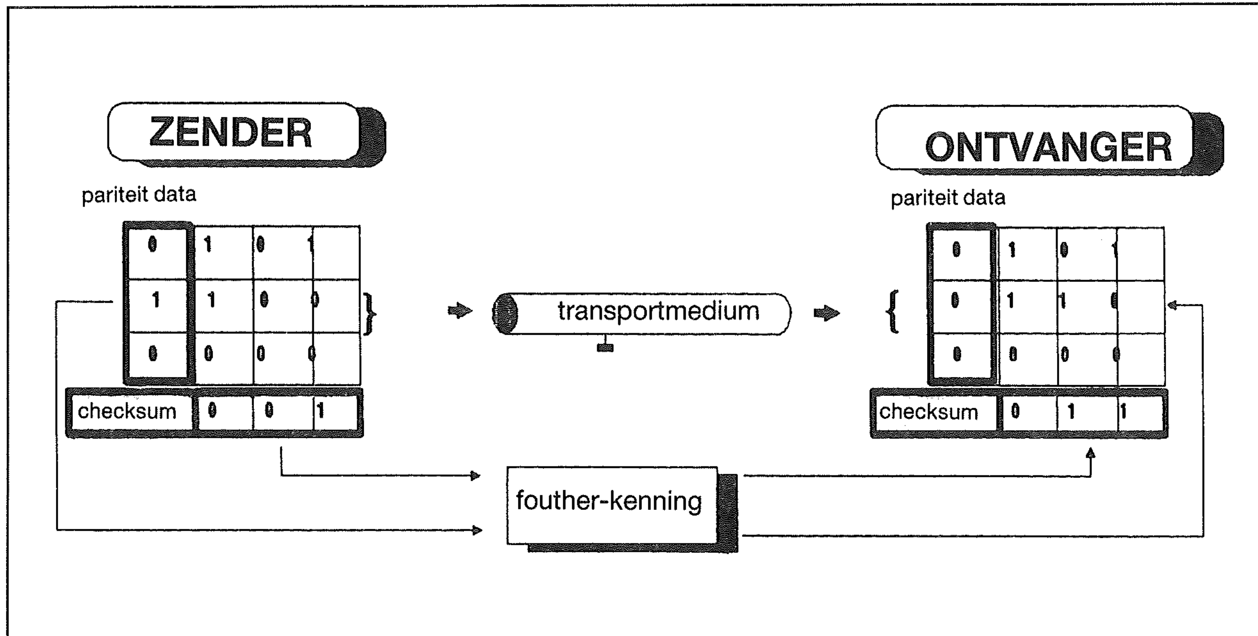
Code-omzetters

In de praktijk is het vaak nodig de ene code in de andere om te zetten. Hiertoe gebruikt men code-omzetters of decoders.

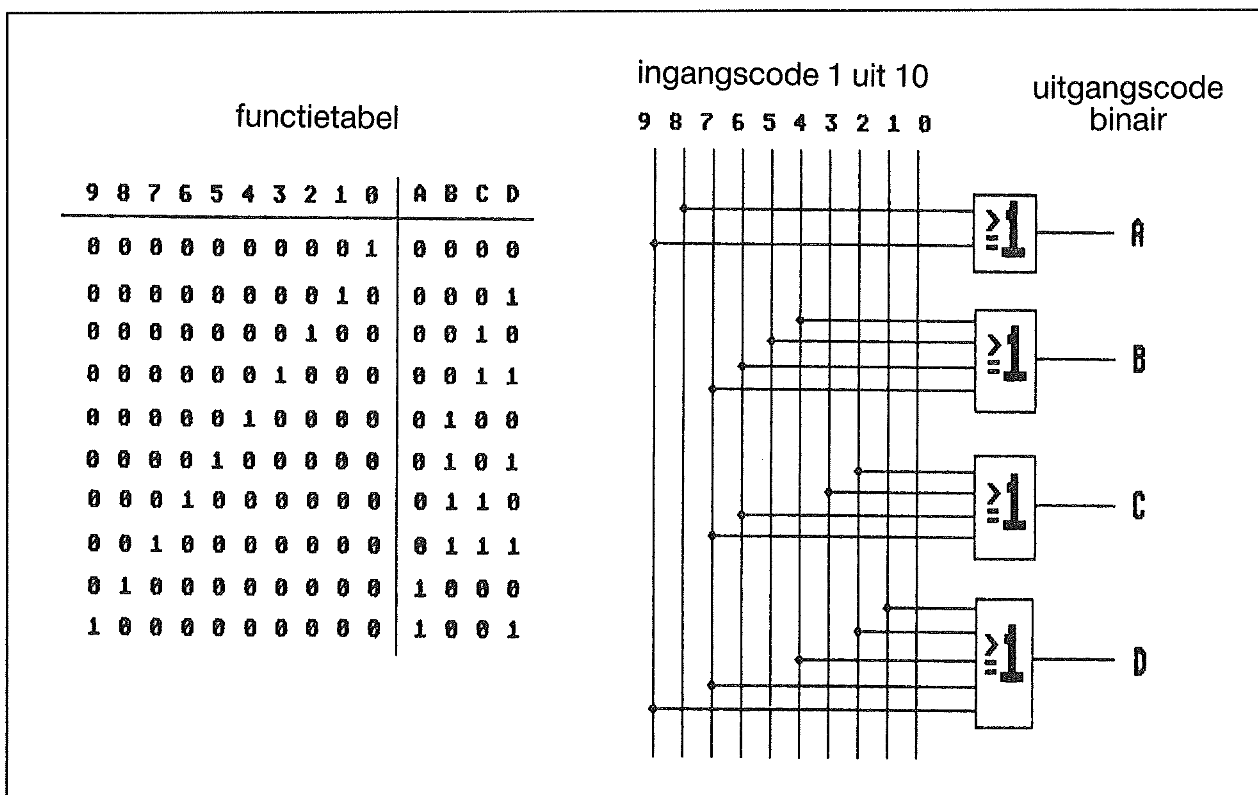
Een code-omzetter die een eenvoudige code in een complexere en meestal compactere code omzet, noemt men een "encoder".
Voorbeeld: de 1 uit 10 naar BCD – encoder van figuur 3/6.8-5.

In het eenvoudigste geval wordt een encoder uit OR-poorten samengesteld. Het aantal OR-poorten is afhankelijk van het aantal gewenste uitgangen. Als er 10 uitgangen nodig zijn, heeft men 10 OR-poorten nodig. Het aantal ingangen van

6.8 Digitale codes en omzetters



Figuur 3/6.8-4: Fout-correctie codes.



9	8	7	6	5	4	3	2	1	0	A	B	C	D
0	0	0	0	0	0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	0	1	0	0	0	0	1
0	0	0	0	0	0	0	1	0	0	0	0	1	0
0	0	0	0	0	0	1	0	0	0	0	0	1	1
0	0	0	0	0	1	0	0	0	0	0	1	0	0
0	0	0	0	1	0	0	0	0	0	0	1	0	1
0	0	0	1	0	0	0	0	0	0	0	1	1	0
0	0	1	0	0	0	0	0	0	0	0	1	1	1
0	1	0	0	0	0	0	0	0	0	1	0	0	0
1	0	0	0	0	0	0	0	0	0	1	0	0	1

Figuur 3/6.8-5: Een encoder.

6.8 Digitale codes en omzetters

de OR-poorten bepaalt hoeveel enen er aan de betreffende uitgang zullen verschijnen.

Een code-omzetter die een complexere en/of compactere code omzet in een eenvoudige code, noemt men "decoder". Bijvoorbeeld de binair naar 1 uit 10 omzetter in figuur 3/6.8-6. Een decoder is samengesteld uit AND-poorten. Voor elke uitgangsvariabele is gewoonlijk een AND-poort nodig.

Een AND-poort heeft zoveel ingangen als er ingangsvariabelen zijn. Natuurlijk kan met Booleaanse algebra de nodige vereenvoudiging worden bereikt.

8421 naar zevensegment decoder

BCD-code				Zevensegment-informatie	segment						
D	B	C	A		a	b	c	d	e	f	g
0	0	0	0	0	1	1	1	1	1	1	0
0	0	0	1	1	0	1	1	0	0	0	0
0	0	1	0	2	1	1	0	1	1	0	1
0	0	1	1	3	1	1	1	1	0	0	1
0	1	0	0	4	0	1	1	0	0	1	1
0	1	0	1	5	1	0	1	1	0	1	1
0	1	1	0	6	1	0	1	1	1	1	1
0	1	1	1	7	1	1	1	0	0	0	0
1	0	0	0	8	1	1	1	1	1	1	1
1	0	0	1	9	1	1	1	1	0	1	1
1	0	1	0	A	1	1	1	0	1	1	1
1	0	1	1	b	0	0	1	1	1	1	1
1	1	0	0	c	0	0	0	1	1	0	1
1	1	0	1	d	0	1	1	1	1	0	1
1	1	1	0	E	1	0	0	1	1	1	1
1	1	1	1	F	1	0	0	0	1	1	1

Verduidelijking: "1" = segment aan

"0" = segment uit

(voor displays met gemeenschappelijke kathode)

Tabel 3/6.8-4: BCD naar zevensegment decoder.

N.B. Een 1 uit n code is een code, waarbij van n uitgangen er altijd slechts een tegelijkertijd log. 1 is.

Een code-omzetter zelf maken

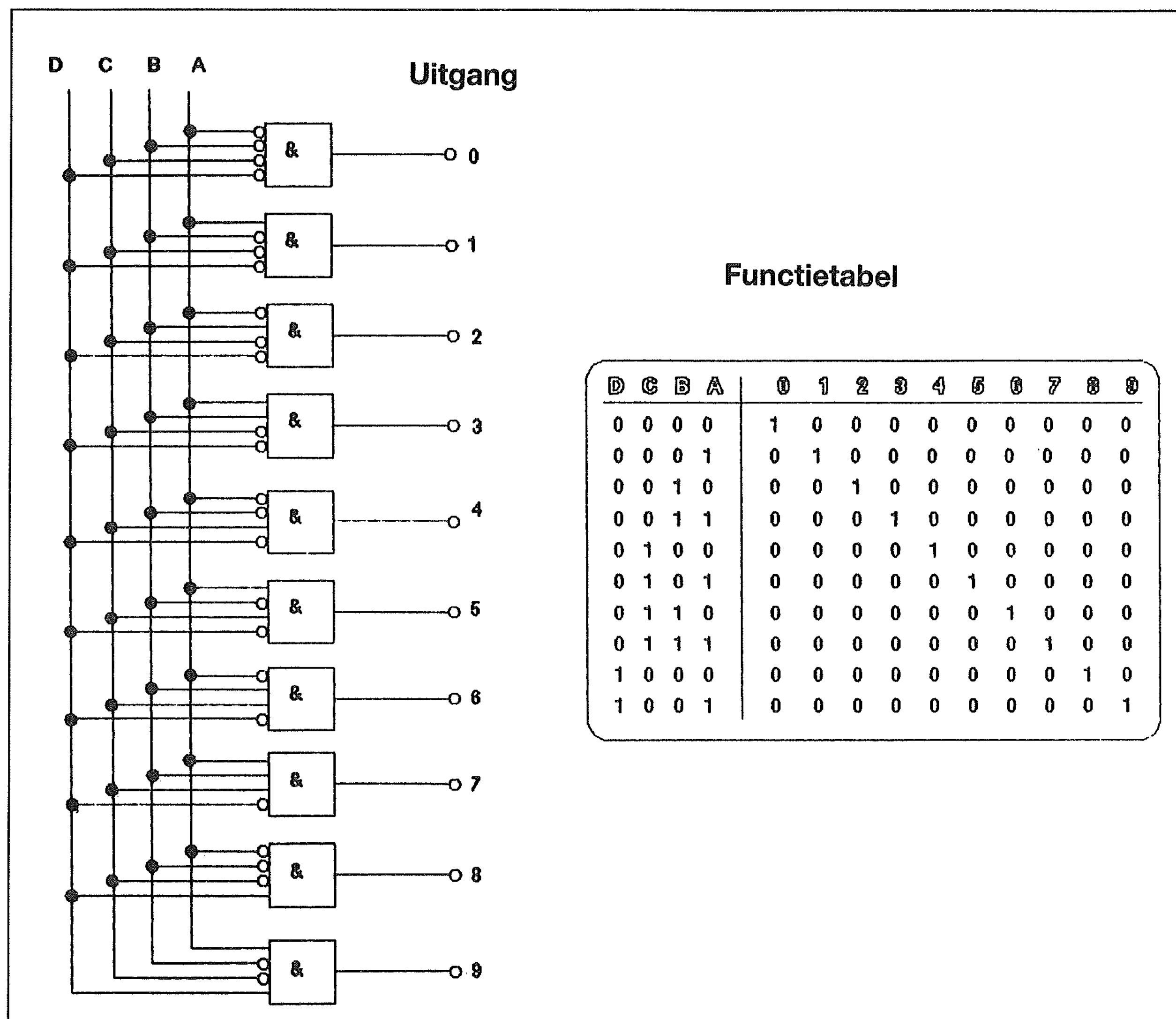
De nu volgende beschrijving laat zien welke stappen men moet doorlopen om een code-omzetter te maken. Als voorbeeld nemen we een code-omzetter, die van een BCD-code een zevensegments-code maakt voor het aansturen van een zevensegments-display. Het bijzondere in dit geval is, dat de code-omzetter alle waarden van 0000 tot 1111 omzet, dus ook de waarden 1000 tot 1111 worden omgezet in de letters A - F.

Eerst bepalen we de gewenste uitgangscode. Elke log. 1 betekent, dat het bijbehorende segment moet oplichten. Wanneer dat gebeurt is afhankelijk van de ingang. De code-omzetter is gemaakt voor zevensegment-displays met een gemeenschappelijke kathode. Zie tabel 3/6.8-4.

De verkregen uitgangscode wordt hierna voor elk segment in een KV-diagram overgenomen. We gebruiken de in eerdere hoofdstukken beschreven technieken om tot een schakeling te geraken met het minimum aantal onderdelen. Zie figuur 3/6.8-7. De max-termen uit de KV-diagrammen zijn samengebracht in tabel 3/6.8-5. Bij het samenstellen van de schakeling kunnen overeenkomstig gebruikte poorten meermaals worden gebruikt. Een standaard TTL-poort kan doorgaans zonder problemen tot 10 TTL-ingangen sturen.

Meestal is het zelf ontwerpen van en- c.q. de-coders niet nodig. Op de markt is een groot aantal IC's verkrijgbaar voor de om-

6.8 Digitale codes en omzetters



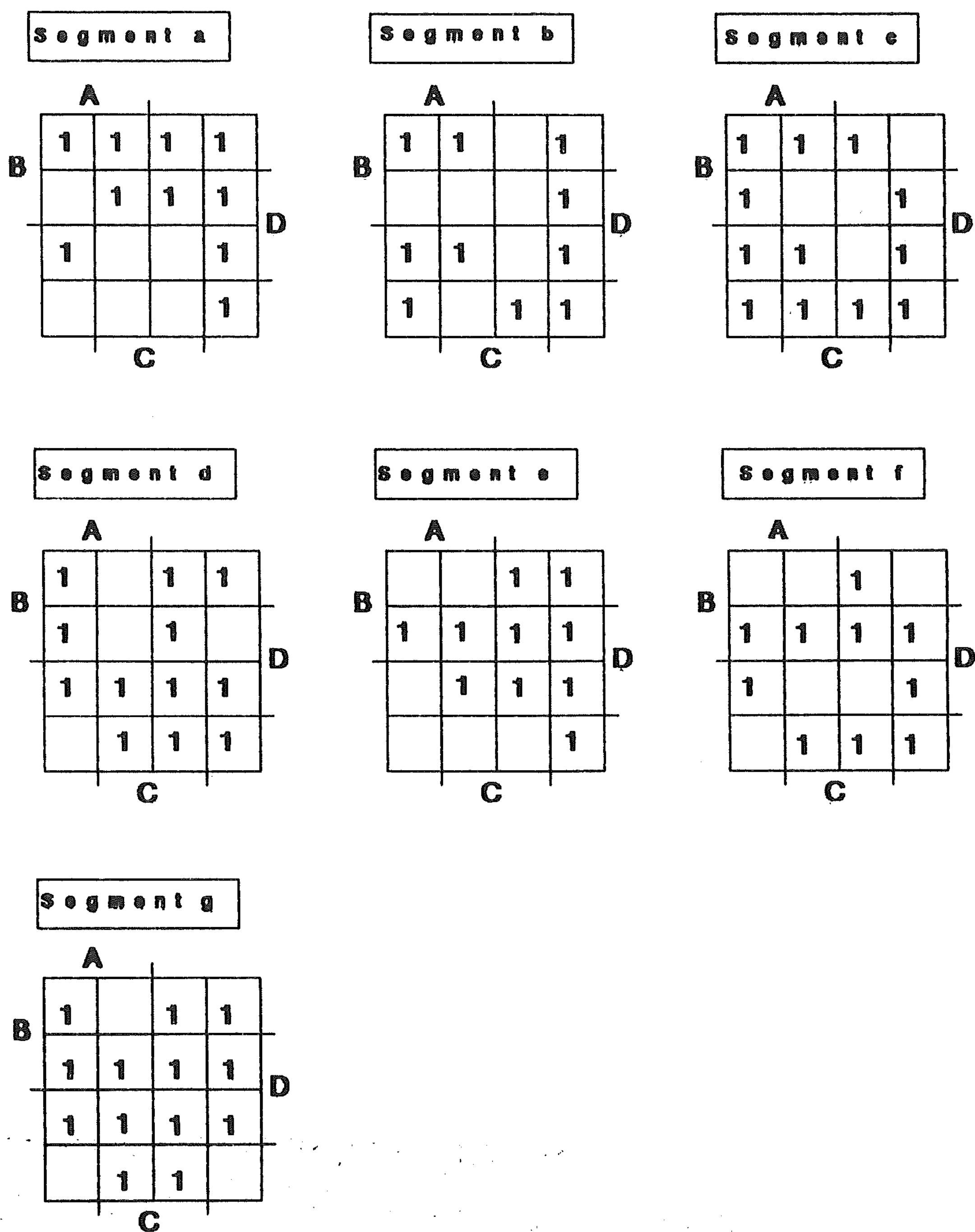
D	C	B	A	0	1	2	3	4	5	6	7	8	9
0	0	0	0	1	0	0	0	0	0	0	0	0	0
0	0	0	1	0	1	0	0	0	0	0	0	0	0
0	0	1	0	0	0	1	0	0	0	0	0	0	0
0	0	1	1	0	0	0	1	0	0	0	0	0	0
0	1	0	0	0	0	0	0	1	0	0	0	0	0
0	1	0	1	0	0	0	0	0	1	0	0	0	0
0	1	1	0	0	0	0	0	0	0	1	0	0	0
0	1	1	1	0	0	0	0	0	0	0	1	0	0
1	0	0	0	0	0	0	0	0	0	0	0	1	0
1	0	0	1	0	0	0	0	0	0	0	0	0	1

Figuur 3/6.8-6: BCD naar 1 uit 10 decoder.

zetting van de meest gangbare codes. Het voordeel ligt niet alleen in de afmeting, maar ook in stroomverbruik en mogelijkheden. Vaak zijn veel voorkomende afwijkingen door middel van het aan de voeding of aan aarde leggen van een ingang nog te kiezen. Denk bij een zevensegments-display bijv. aan de zes wel of niet met een horizontaal streepje bovenaan. Een voorbeeld van een standaard code-omzetter vindt u in figuur 3/6.8-9.

De digitale codes vormen de basis voor het begrijpen van digitale tellers. Deze komen in een volgend hoofdstuk aan de orde.

6.8 Digitale codes en omzetters



Figuur 3/6.8-7: De uitgangsvergelijkingen in de KV-diagrammen.

6.8 Digitale codes en omzetters

Segment :		MAX TERM
a	=	$(\overset{\downarrow}{B} * \overline{D}) + (\overline{A} * \overline{C}) + (B * C) + (\overline{B} * \overline{C} * D)$
b	=	$(\overline{A} * \overline{C}) + (A * B * \overline{D}) + (A * \overline{B} * D) + (\overline{B} * \overline{C} * \overline{D}) + (\overline{A} * \overline{B} * \overline{D})$
c	=	$(A * \overline{C}) + (\overline{B} * \overline{D}) + (A * \overline{B}) + (\overline{C} * D) + (C * \overline{D})$
d	=	$(\overline{B} * D) + (A * \overline{B} * C) + (\overline{A} * \overline{C} * \overline{D}) + (\overline{A} * B * C) + (A * B * C)$
e	=	$(\overline{A} * \overline{C}) + (B * D) + (C * D) + (\overline{A} * B)$
f	=	$(\overset{\downarrow}{B} * D) + (\overline{A} * \overline{B} * \overline{C}) + (\overline{B} * C * \overline{D}) + (A * \overline{C} * D) + (\overline{A} * B * C)$
g	=	$D + (\overline{B} * C) + (\overline{A} * B) + (B * \overline{C})$

waar:

+ = logische OR-functie

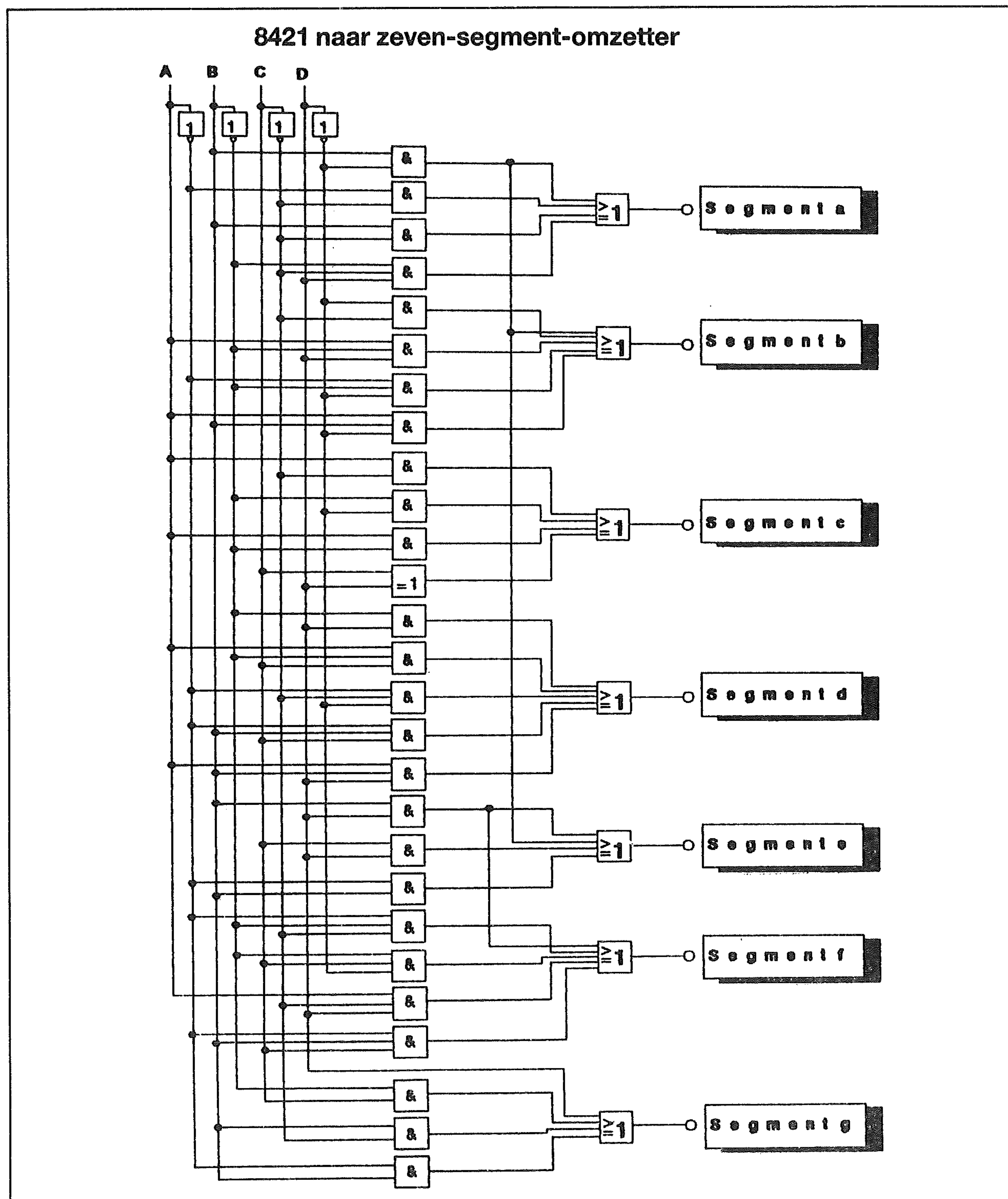
$\downarrow$ = deze term wordt meer dan een maal gebruikt

* = logische AND-functie

- = inverse waarde

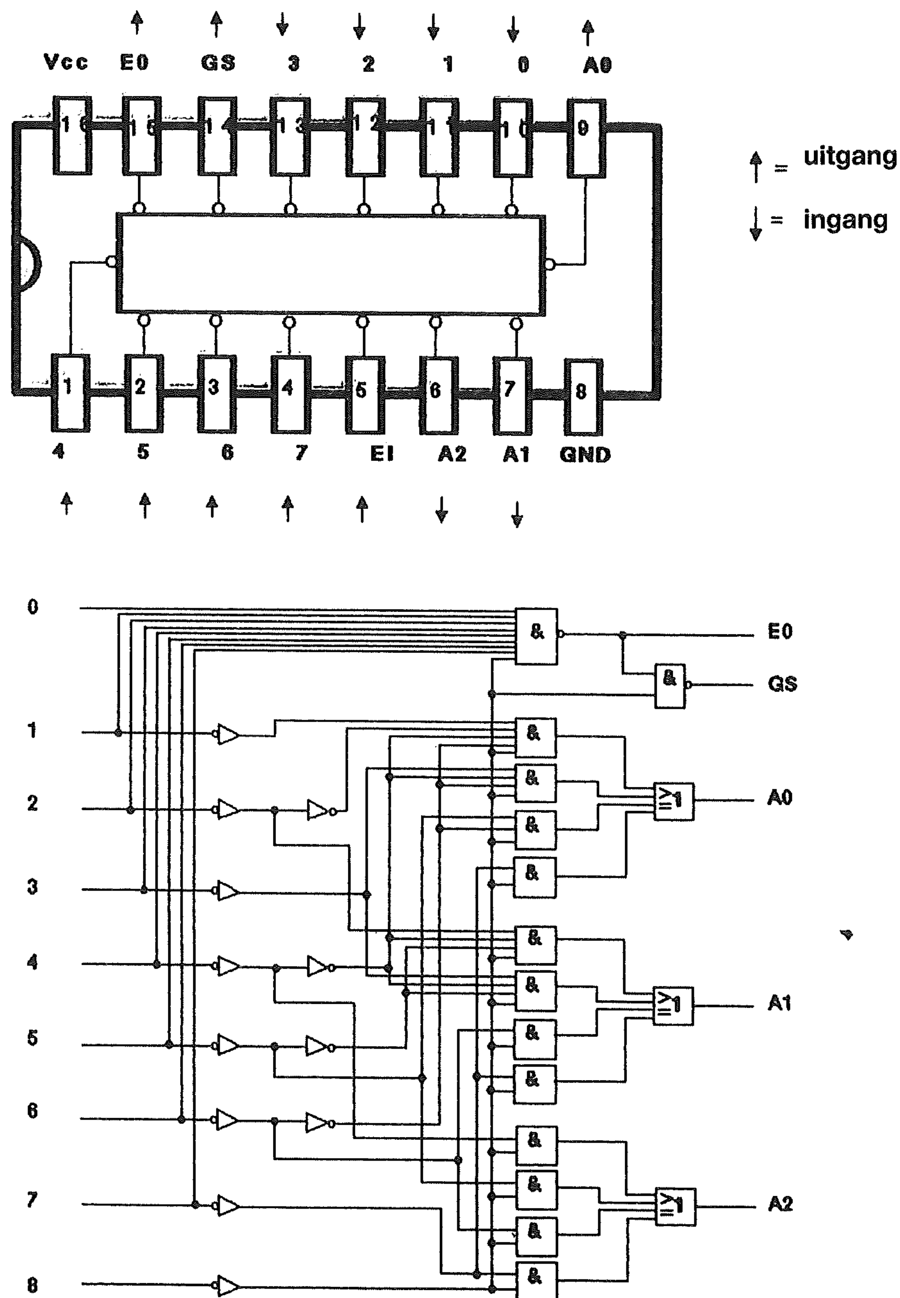
Tabel 3/6.8-5: De max-termen uit de KV-diagrammen.

6.8 Digitale codes en omzetters



Figuur 3/6.8-8: De schakeling van een code-omzetter.

6.8 Digitale codes en omzetters



Figuur 3/6.8-9: Een 74146 code-omzetter met schema.

3/6.9

Digitale telschakelingen

Tellen wil eigenlijk zeggen een bepaalde meetwaarde of meer algemeen een bepaalde ingangsgrootheid uitdrukken in een getal. In de digitale techniek is een getalswaarde zélfs de enig mogelijke voorstelling van een grootheid.

Als voorbeeld halen we een frequentieteller aan. Deze geeft door een getalvoorstelling aan, hoe vaak het aangeboden signaal binnen een bepaalde gekozen periode van spanningsniveau verandert. Laten we deze funktie eens aan een wat nadere beschouwing onderwerpen. We maken daarbij gebruik van het blokschema van figuur 3/6.9-1, waarin een 4-digit frequentieteller is getekend.

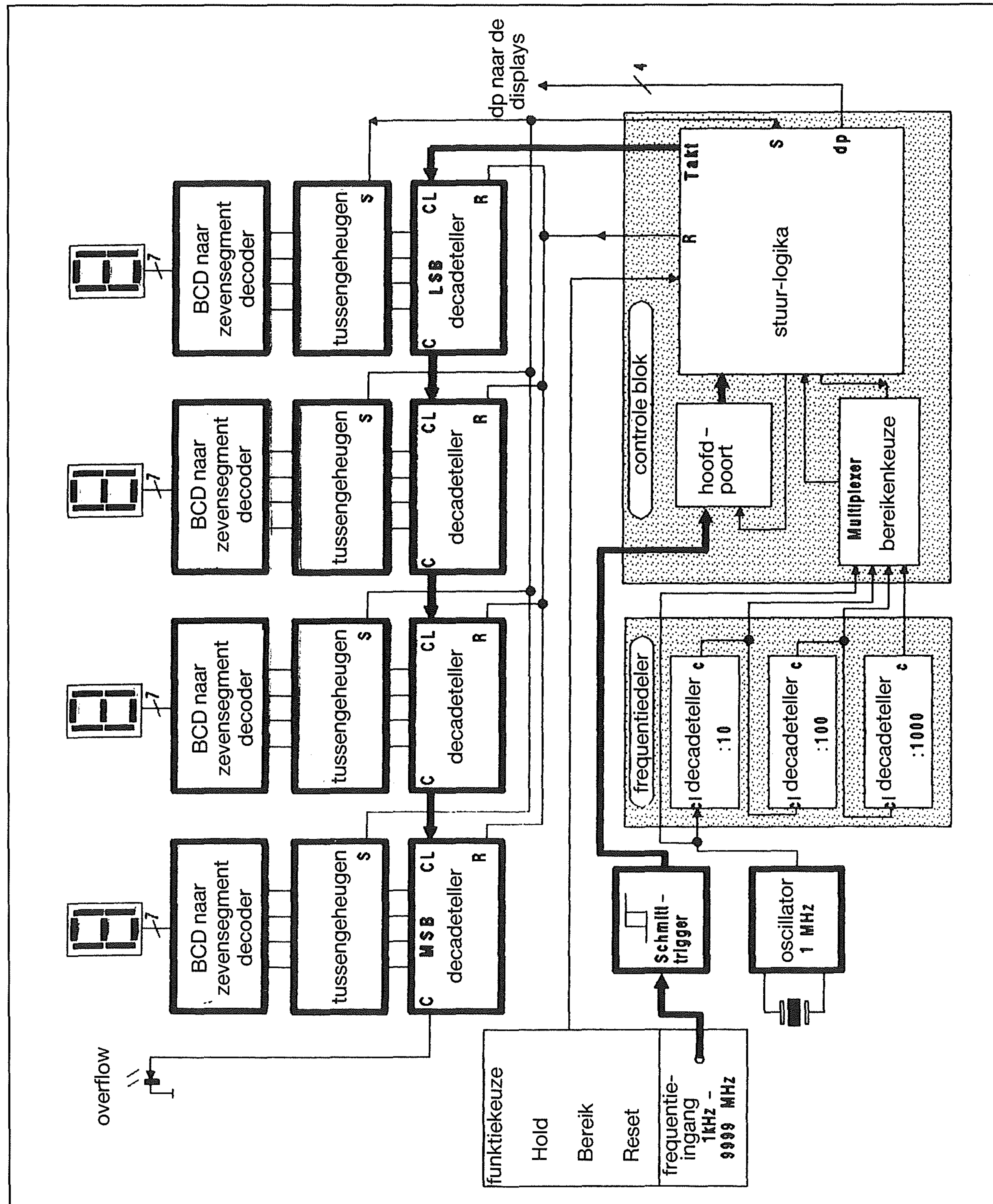
De oscillator levert een kristalfrequentie aan de hoofdpoort. Hoe lang de hoofdpoort open staat is afhankelijk van de periodeduur van de door het kristal geleverde referentie frequentie. Als de tijd dat de poort open staat bijvoorbeeld 1 ms bedraagt, dan wordt de frequentie van het aangeboden signaal weergegeven in kHz. Als de oscillatorfrequentie afneemt, wordt de periode langer, waardoor meer ingangspulsen de hoofdpoort kunnen passeren. Dat wil zeggen dat de aangeduide getalswaarde zal toenemen. De oscillatorfrequentie wordt door de decade-tellers in het frequentiedeler blok gedeeld. In dit voorbeeld deelt de eerste trap door 10, de tweede door 100 en de derde door 1000. Afhankelijk van het gekozen bereik, wordt de uitgangsfrequentie van de eerste, de tweede of de derde trap via de multiplexer aan de hoofdpoort en de stuurtrap aangeboden.

De multiplexer werkt als een keuze schakelaar, die steeds de uitgang verbindt met maar een van de ingangen. Als de overflow indicator aangaat, is het gekozen meetbereik te klein. De tijd dat de hoofdpoort openstaat is te lang. Er worden meer signaalwisselingen bij het gekozen bereik gemeten dan het display kan aangeven. Men zal dus een ander bereik moeten kiezen.

Welke decimale punt (dp) aangestuurd wordt, is afhankelijk van het gekozen bereik. De stuurlogika zorgt voor de keuze van de juiste decimale punt.

De ingangsfrequentie wordt eerst aan een Schmitt-trigger aangeboden, die van een langzaam veranderend (analoog) signaal een pulsvormig (digitaal) signaal maakt. Het ingangssignaal wordt vaak met voorversterkers op niveau gebracht en eventueel met filters van ongewenste parasieten ontdaan. Ook de trigger niveaus van de Schmitt-trigger zijn vaak in te stellen, zodat elk ingangssignaal te verwerken is. Het gedigitaliseerde ingangssignaal wordt aan de hoofdpoort aangeboden.

6.9 Digitale telschakelingen



Figuur 3/6.9-1: Blokschema van een frequentiemeter met 4 digits.

6.9 Digitale telschakelingen

Onder hoofdpoort verstaan we in dit voorbeeld de AND-poort die de ingangspulsen die van de Schmitt-trigger komen blokkeert of doorlaat naar de rest van de logika. Daartoe wordt het ingangssignaal aan de ene ingang aangeboden en het stuursignaal aan de andere.

Het tijddiagram van figuur 3/6.9-2 geeft een goed overzicht van een totale meetcyclus.

Voordat de hoofdpoort wordt vrijgegeven zorgt een resetpuls van de stuurlogika ervoor, dat alles (behalve de tussengeheugens) op nul worden gezet (tijdstip I). De decadetellers staan dus op de waarde 0 [0000(BCD)]. Afhankelijk van het gekozen bereik bepaalt de stuurlogika hoe lang de hoofdpoort wordt vrijgegeven en dus hoe lang de ingangspulsen vrijelijk naar de klok-ingang (CL) van de LSD-decade teller kunnen (LSD =Least Significant Digit = cijfer met de laagste waarde). Tijdstip II.

Deze decadeteller begint nu de ingangspulsen te tellen. Elke decadeteller kan tot negen [1001(BCD)] tellen, dan begint hij opnieuw bij nul. Als een teller aan zijn maximale stand is gekomen, dan verschijnt er een korte puls op de Carry-uitgang (Carry = overdracht). Tijdstip III. Hiermee wordt de volgende decadeteller geklokt.

Als de meetduur is afgelopen, dan wordt de hoofdpoort weer geblokkeerd. Daarmee is het tellen onderbroken. Tijdstip IV. De waarde van de decadetellers wordt in BCD-vorm aan de tussengeheugens aangeboden. Kort na het sluiten van de hoofdpoort genereert de stuurlogika een 'store' (overname) puls. Tijdstip V. De waarde in de decadetellers wordt nu overgenomen in de tussengeheugens. Via de BCD naar zevensegment-decoders wordt de gemeten waarde op de displays zichtbaar.

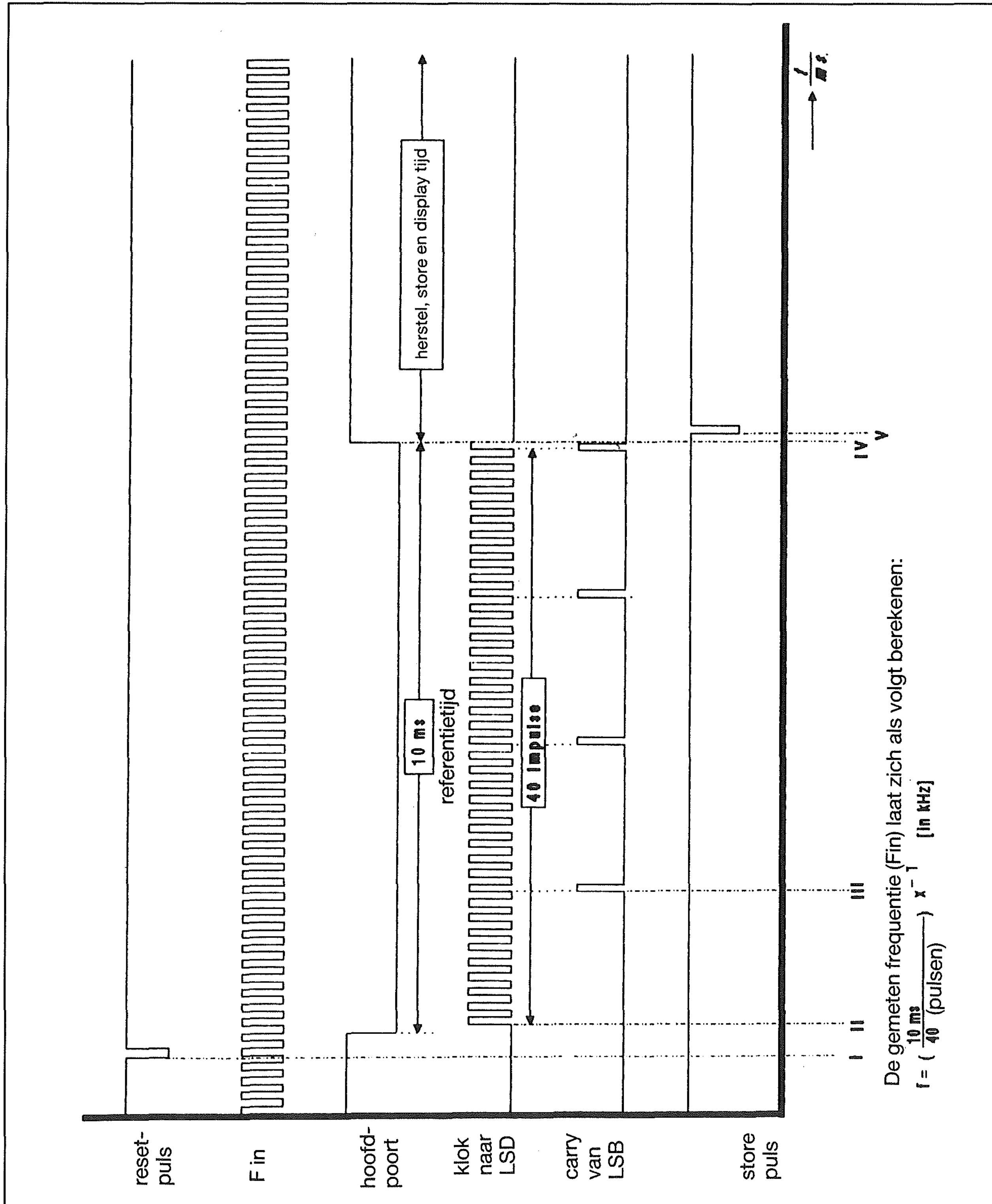
De tussengeheugens zorgen ervoor, dat de displays tijdens de meetcyclus niet voortdurend van waarde veranderen. Zonder tussengeheugens zou het niet mogelijk zijn de displays af te lezen. Door het snelle doorlopen van de tellers en de traagheid van het menselijke oog zou het lijken, of alle segementen steeds aan zijn, waardoor we voortdurend een acht zouden zien.
Na een bepaalde hersteltijd, wordt de cyclus automatisch herhaald. De hold funktie 'bevriest' het display, dat wil zeggen dat zelfs al verandert de ingangsfrequentie, het display gelijk blijft. Met de Reset worden de tellers weer op nul gezet.

Decadetellers zijn een van de soorten tellers, die we bij digitale telschakelingen tegenkomen. Hoe decadetellers worden samengesteld, welke soorten tellers er nog meer zijn en hoe men zelf tellers met een bepaalde gewenste funktie kan ontwerpen wordt in de rest van dit hoofdstuk uit de doeken gedaan.

Waaruit bestaan telschakelingen?

Een telschakeling bestaat in het algemeen uit een aaneenschakeling van geklokte flip-flops. In sommige teller IC's kunnen we met de huidige integratie dichtheid zeer vele flip-flops in één IC worden ondergebracht. Het resultaat is zeer complexe telschakelingen in een IC. De manier waarop de flip-flops en hun ingangen onderling zijn verbonden bepaalt de funktie van de telschakeling. In het algemeen verandert de telschakeling bij elke geldige ingangspuls op de klok een of meerdere uit-

6.9 Digitale telschakelingen



Figuur 3/6.9-2: Tijddiagram van een meetcyclus in de frequentieteller.

6.9 Digitale telschakelingen

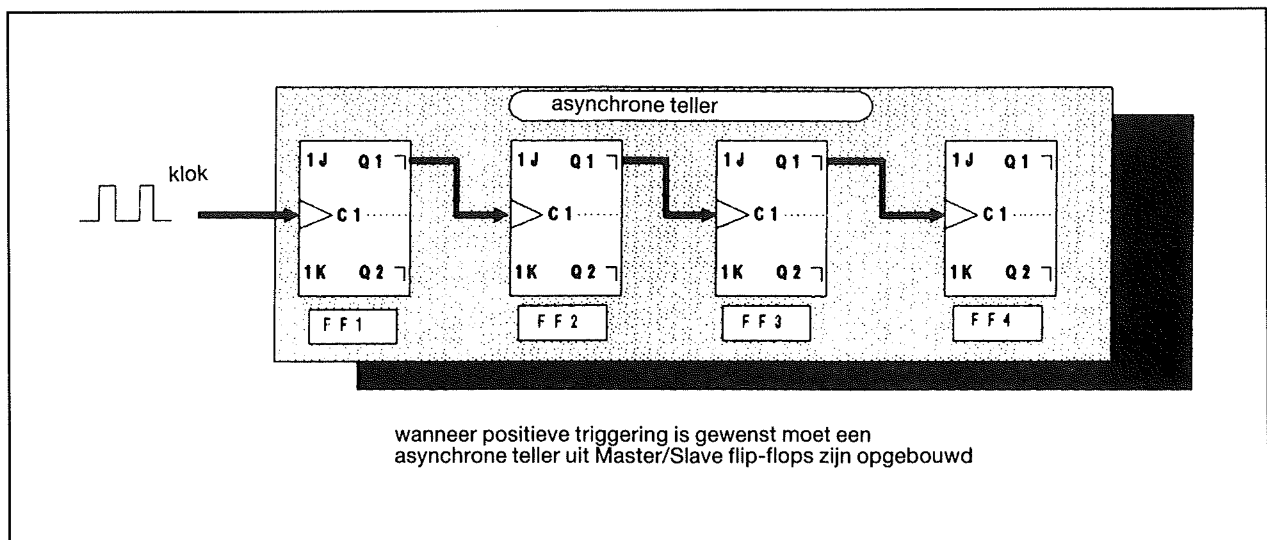
gangstoestanden. Doorgaans hebben we te maken met een oplopende of dalende waarde.

Als de (gedecodeerde) waarde van de uitgangen toeneemt bij elke klokpuls, dan spreken we van voorwaartstellers. Wanneer daarintegen de waarde daalt spreekt men van achterwaarts of terugtellers. De engelse termen 'forward counter' en 'reverse counter' duiden respectievelijk een voorwaarts en achterwaarts teller aan. De tellers kunnen verschillende soorten uitgangscodes genereren zoals BCD-code, Aiken-code, Excess-code, etc. Deze codes zijn al eerder behandeld.

In de digitale schakeltechniek geldt de toestand waarin alle uitgangen laag zijn als reset-waarde. Een tienteller (of decade-teller) zal bijvoorbeeld niet van 1 tot 10, maar van 0 tot 9 tellen. Met tientaller wordt dus bedoeld, dat de teller 10 verschillende uitgangswaarden kent. Bij de volgende behandeling, maar zeker ook in de praktijk moet u dit goed in gedachten houden!

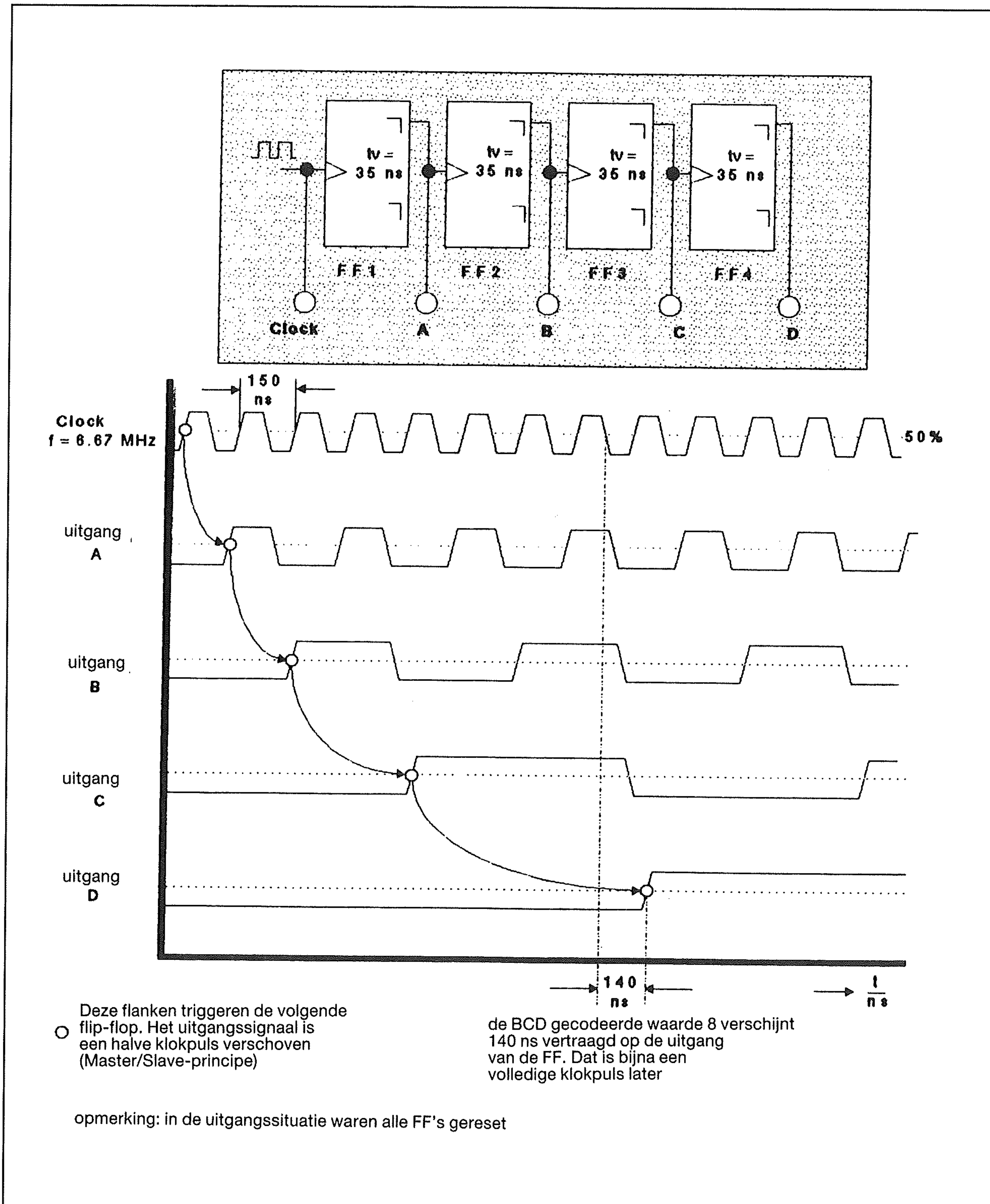
Het aantal gebruikte flip-flops is bepalend voor de maximale tellerstand. Elke flipflop kan twee waarden aannemen. Logisch 1 en logisch 0. Als in een tellerschakeling n flip-flops voorkomen, dan kan de maximale tellerstand met de volgende formule worden berekend: $W_{(max)} = 2^n-1$. Dus een telschakeling met vier flip-flops kent $2^4 = 16$ tellerstanden. De maximale waarde is $2^4-1 = 15$. De mogelijke waarden zijn 0-15. De maximale waarde kán voorgesteld worden. Het is echter ook mogelijk dat de teller via poorten en terugkoppelingen wordt gereset voordat de maximale teller stand is bereikt.

Bij de algemene beschrijving van tellers en hun uitgangstoestanden wordt vaak het begrip 'modulo-n-teller' gebruikt. Hiermee wordt bedoeld het door n aangegeven aantal mogelijke uitgangstoestanden. Een decade (= tien) teller kan worden aangeduid als een modulo-10-teller.



Figuur 3/6.9-3: Asynchrone teller.

6.9 Digitale telschakelingen



Figuur 3/6.9-4: De limieten van asynchrone tellers.

De gebruikte flip-flops

In tellers worden nagenoeg altijd flankgetriggerde flip-flops gebruikt om een grote storingsongevoeligheid te verkrijgen. Bij deze flip-flops verandert de uitgang pas nadat de ingangen stabiel zijn en op de klokingang een positieve flank is verschenen (niveauverandering van logisch 0 naar logisch 1). Ook triggering op de negatieve flank komt voor.
Hoe complex de schakeling aan de ingangen wordt is afhankelijk van de manier van klokken. Bij sommige tellers kunnen alleen Master/Slave flip-flops worden gebruikt. Bij Master/Slave flip-flops is het uitgangssignaal een halve klokpuls verschoven ten opzichte van de klok.

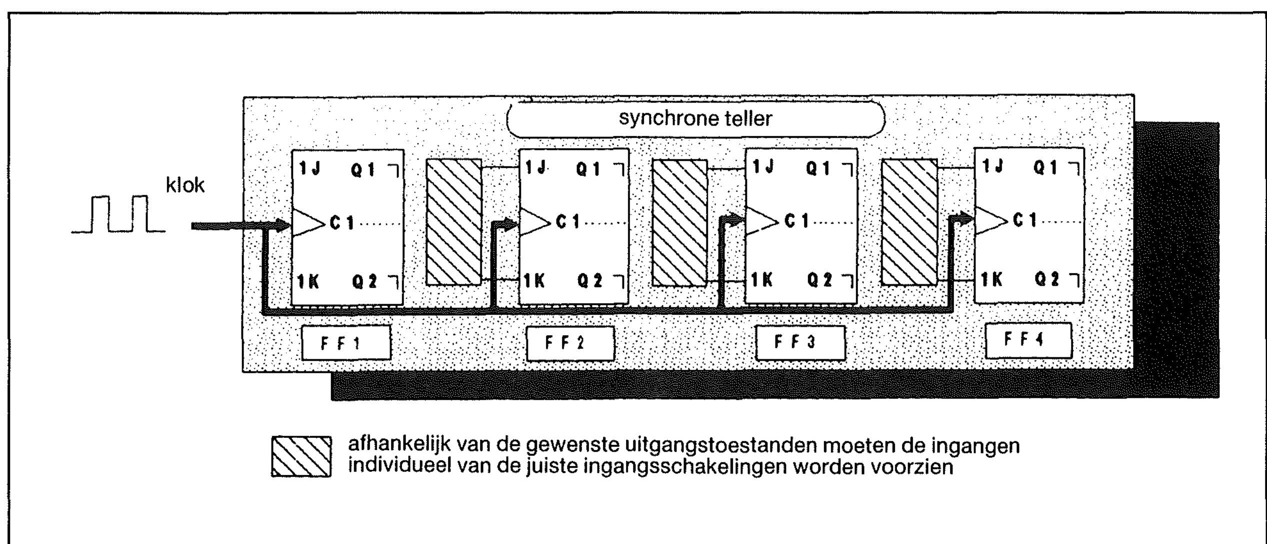
De verschillende tellersoorten

Naar gelang de manier van klokken worden tellers in twee hoofdgroepen ingedeeld.

1. Asynchrone tellers

Bij deze tellers komt het kloksignaal op de eerste flip-flop, zie figuur 3/6.9-3. Alle volgende flip-flops worden geklokt met het uitgangssignaal van hun voorganger.

Het voordeel van asynchrone tellers is dat er geen ingewikkelde logische schakelingen voor de ingangen nodig zijn. Een nadeel is de vertraging die het signaal oploopt in elke poort, de zogenaamde propagation delay. Dit is de tijd die verloopt tussen een signaalverandering op de ingang en de verandering tengevolge hiervan op de uitgang. Bij standaard TTL-poorten bedraagt deze vertraging voor één flip-flop 25-50 ns. C-MOS flip-flops zijn wat minder snel. De vertraging ligt daarbij tussen 35-70 ns. Bij een schakeling met vier flip-flops achter elkaar geschakeld komt de signaalverandering op de uitgang van de laatste flip-flop 4 x 35 = 140 ns na de verandering op de ingang. In figuur 3/6.9-4 zien we wat de gevolgen zijn!
Deze eigenschap van asynchrone tellers beperkt de maximale frequentie waarbij



Figuur 3/6.9-5: Synchrone teller.

zij nog bruikbaar zijn drastisch. De maximale frequentie van een asynchrone telschakeling is bereikt, als de totale propagation delay groter of gelijk wordt aan de periodeduur van de klok. Als men een hogere klokfrequentie gebruikt, is de uitgangscode niet meer correct. In de praktijk zal tengevolge van de spreiding in propagation delay van de gebruikt IC's zelfs een ruimere marge nodig zijn en moet men zorgen dat de totale maximum te verwachten propagation delay korter is dan de periodeduur van de klok.

2. Synchrone tellers

In tegenstelling tot de asynchrone tellers, wordt bij synchrone tellers het kloksignaal tegelijkertijd aan alle flip-flops van de schakeling aangeboden. In figuur 3/6.9-5 zien we het principeschema van een synchrone teller. Alle flip-flops nemen tegelijkertijd de nieuwe uitgangspositie aan (kleine afwijkingen in onderlinge propagation delays daargelaten). De vertraging kan hooguit nog de vertragingstijd van de traagste flip-flop in de schakeling zijn.

Omdat alle flip-flops tegelijkertijd worden geklokt, moeten de ingangen van de verschillende flip-flops voor de klokpuls op de volgende verandering worden voorbereid.
Hiervoor worden de uitgangssignalen van de gebruikte flip-flops gebruikt. De ontwerper moet dus goed op de hoogte zijn van de waarheidstabel van de individuele flip-flops. De complexiteit van het ontwerp neemt toe. Het wordt pas echt ingewikkeld, als een synchrone teller zowel voor- als achteruit moet kunnen tellen.

Synchrone tellers kunnen worden gebruikt met klokfrequenties die boven de 20 MHz liggen!

Tellerschakelingen nauwkeuriger bekeken

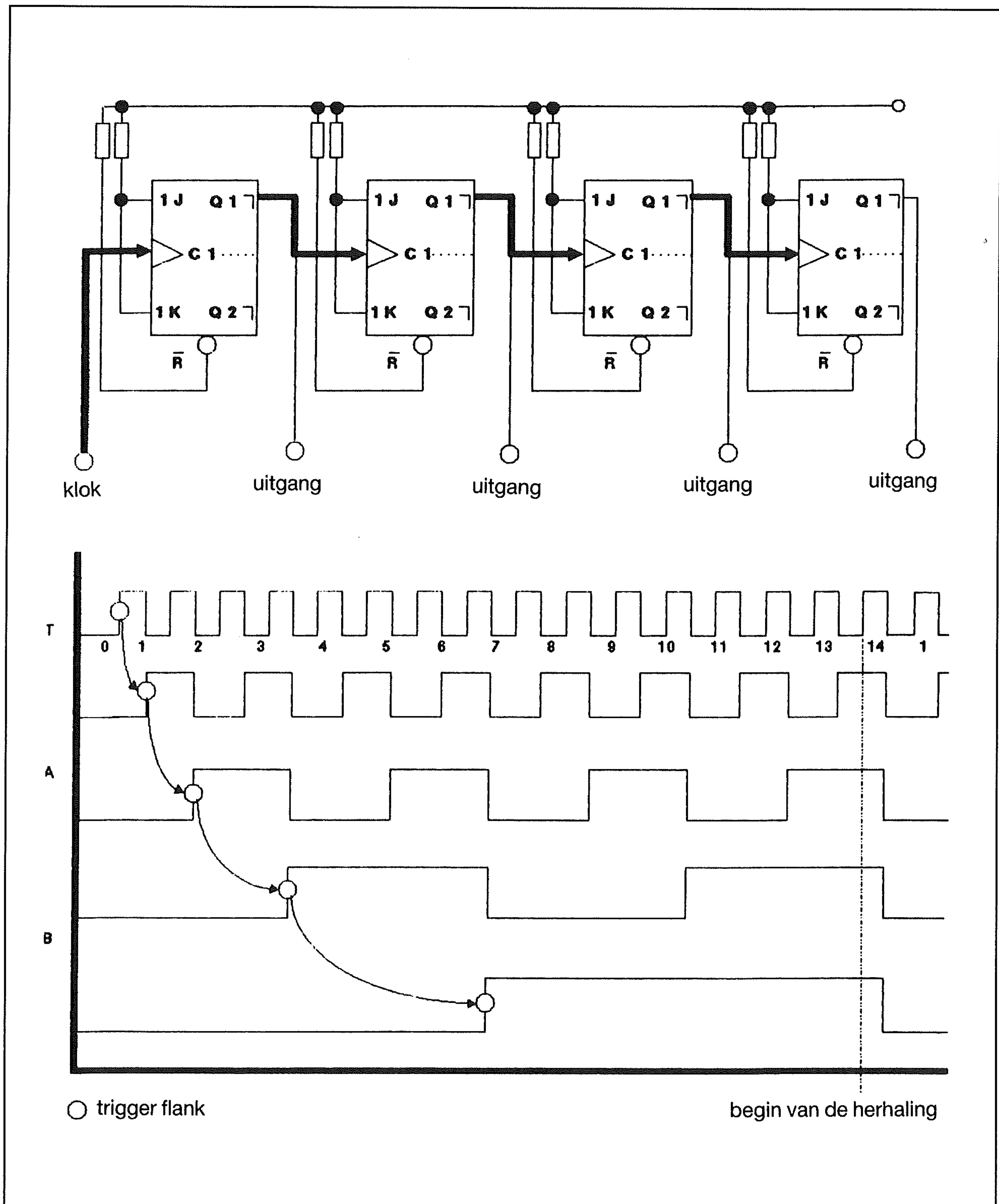
Bij de nu volgende tellers wordt de funktie doorgaans beschreven aan de hand van de tijddiagrammen en/of funktietabellen. U zult zien dat de plaatjes vele malen meer zeggen dan tekst. Wat gebeurt en wanneer dat gebeurt, alsmede kritische tijdstippen zijn aan de hand van tijddiagrammen eenvoudig te volgen. Voor elk type telschakeling wordt zo mogelijk een uitleg gegeven met verschillende typen flipflops, waarbij de voor- en nadelen van toepassing van de gebruikte typen naar voren zullen komen.

1. De asynchrone binaire voorwaartsteller

De funktie van deze teller is zo eenvoudig, dat een verdere uitleg niet nodig is. De teller bestaat uit flip-flops, die als frequentiedelers achter elkaar zijn geschakeld. Iedere flip-flop deelt de klokfrequentie door twee.

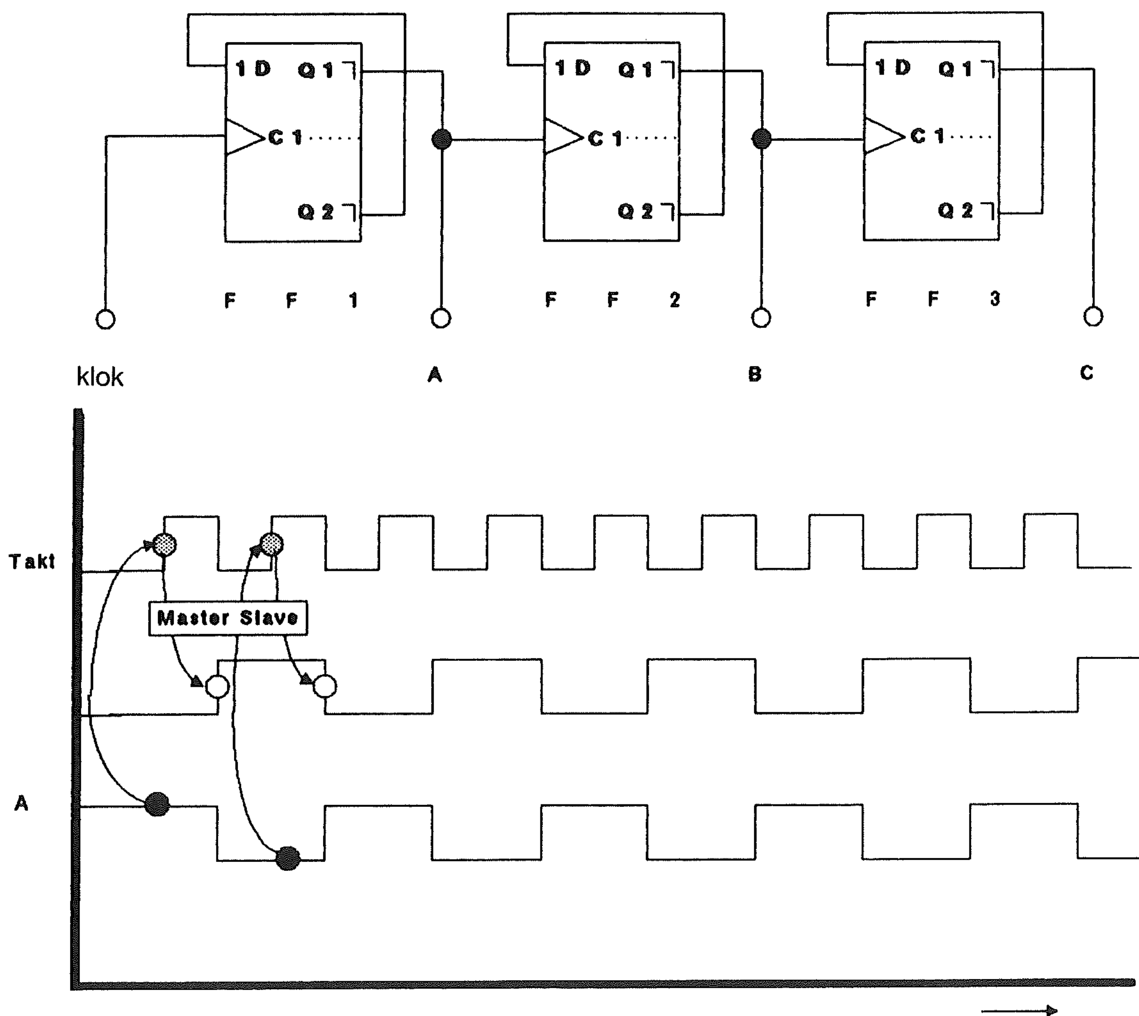
Als de triggering op de positieve flank van de klok moet gebeuren kunnen alleen Master/Slave flip-flops worden gebruikt. Het best zijn dan de JK-Master/Slave flip-flops, omdat die geen andere stuursignalen nodig hebben (zoals bijvoorbeeld R-S flip-flops). Zie figuur 3/6.9-6 voor een binaire voorwaartsteller met JK-Master/Slave flip-flops.
Wil men de schakeling realiseren met D-flip-flops (Master/Slave), dan moeten de nodige doorverbindingen worden gelegd om een 'toggle' situatie te creëren. Toggle wil zeggen, dat de flip-flops, zoals van een flip-flop te verwachten is, zijn uitgang na iedere klokpuls inverteert. Daartoe verbinden we de inverterende uitgang van elke flip-flop met zijn D-ingang. Zie figuur 3/6.9-7 voor de uitwerking.

6.9 Digitale telschakelingen



Figuur 3/6.9-6: Voorwaartsteller met JK-Master/Slave flip-flops.

6.9 Digitale telschakelingen



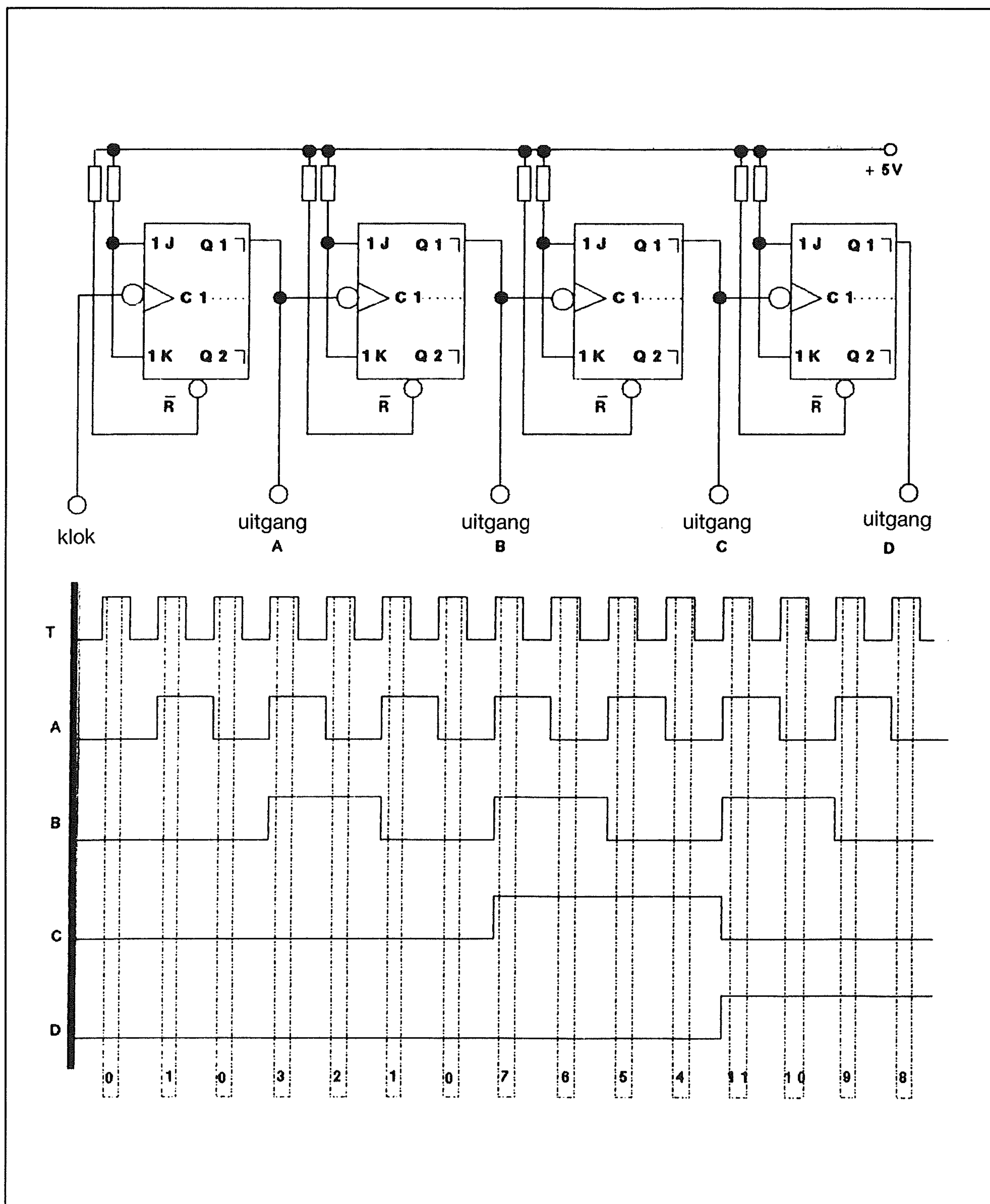
de logische toestand op Q2 zorgt voor het juiste niveau op de D-ingang voor de volgende wissel, dit veroorzaakt de toggle mode

op de positieve flank van de klok neemt de D-FF de ingangsinformatie over, het Master/Slave principe zorgt voor de vertraging

de telvolgorde komt overeen met die van figuur 3/6.9-6

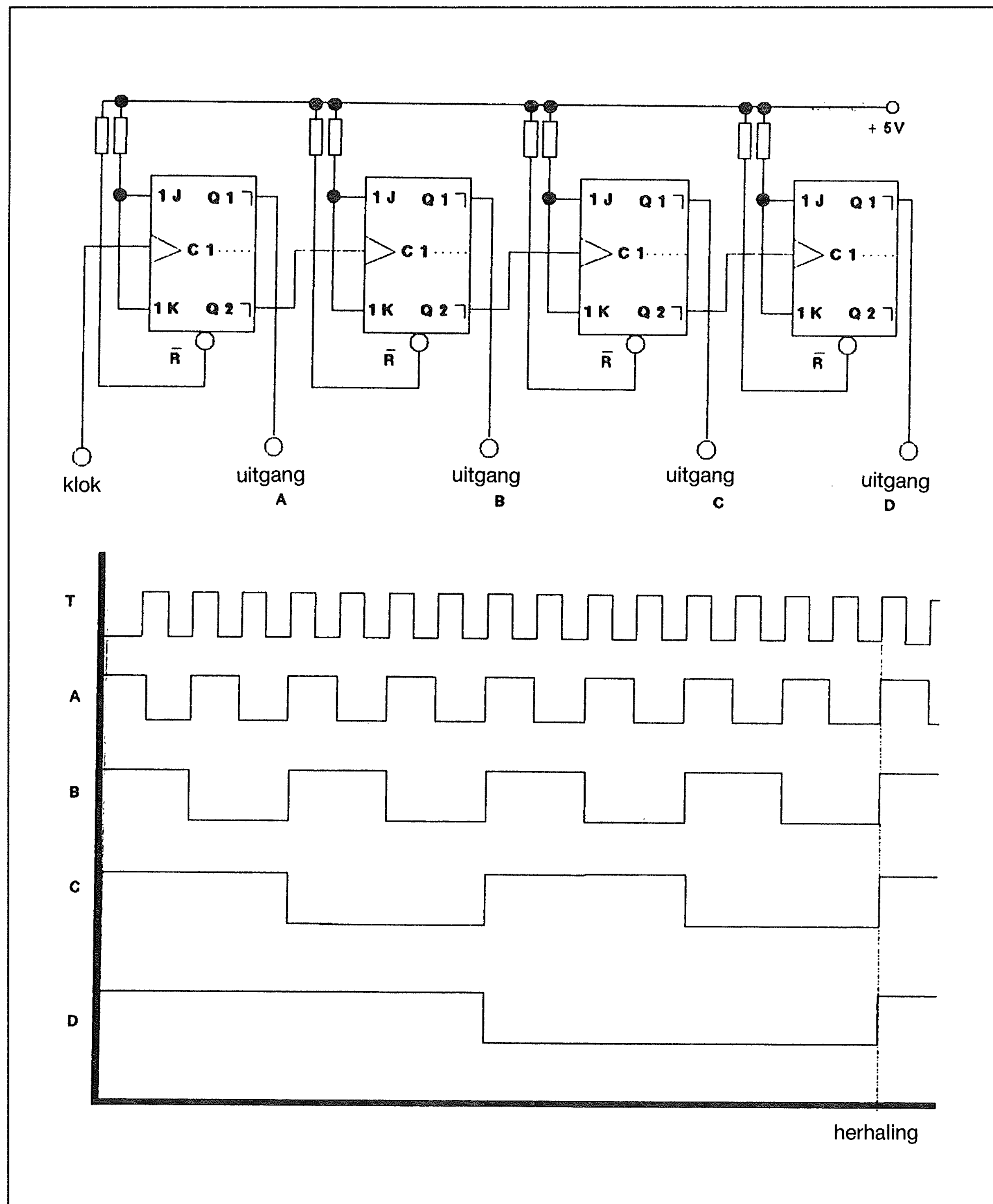
Figuur 3/6.9-7: D-Master/Slave flip-flops in een toggle mode.

6.9 Digitale telschakelingen



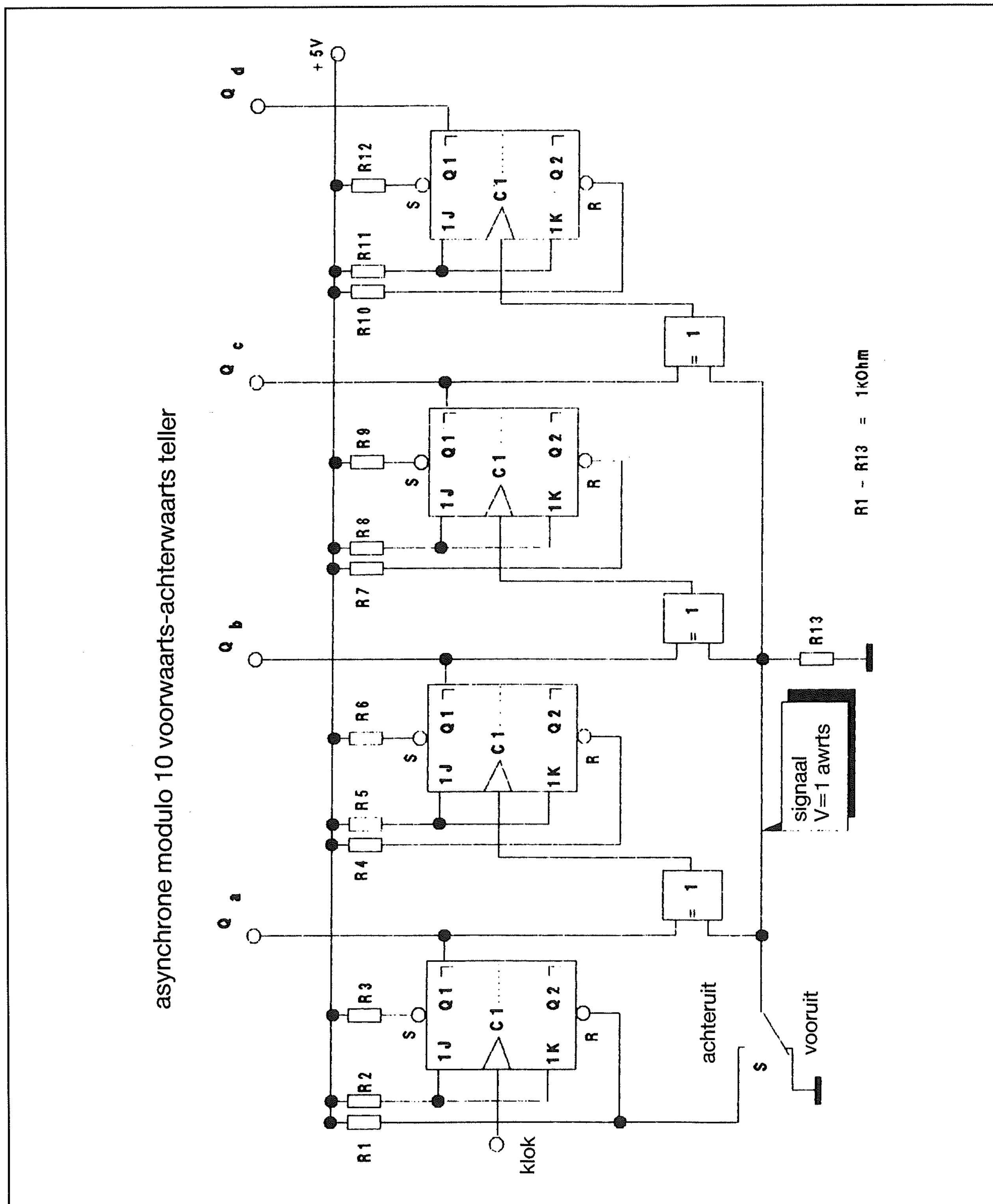
Figuur 3/6.9-8: Het foutieve telresultaat van een teller met Master/Slave FF's bij neg. flank-triggering.

6.9 Digitale telschakelingen



Figuur 3/6.9-9: Binaire terugteller met JK-Master/Slave flip-flops met positieve flanktriggering.

6.9 Digitale telschakelingen



Figuur 3/6.9-10: Asynchronische voorwaarts/achterwaartsteller.

6.9 Digitale telschakelingen

Als de sturing op de negatieve flanken van de klok moet gebeuren, dan kunnen geen Master/Slave flip-flops worden gebruikt. Bij gebruik van een Master/Slave flip-flop zou binair tellen uitgesloten zijn. De flip-flops zouden het ingangssignaal namelijk een volledige klokpuls vertragen. In figuur 3/6.9-7 is dit duidelijk uit het tijddiagram te destilleren. In figuur 3/6.9-8 is te zien wat het uitgangsresultaat is van een asynchrone teller met JK-Master/Slave flip-flop, die op de negatieve flank worden getriggerd.

2. De asynchrone binaire achterwaartsteller

Een achterwaartsteller, of terugteller, zal bij elke klokpuls zijn uitgangswaarde verlagen. De teller begint bij de hoogste waarde en telt dan terug, totdat de laagste waarde is bereikt. Daarna wordt de cyclus herhaald.
In de praktijk bereikt men dit, door bijvoorbeeld de volgende flip-flop niet te klokken met de gewone uitgang van de vorige flip-flop, maar met de geïnverteerde uitgang, meestal Q2 genoemd. In figuur 3/6.9-9 vinden we een binaire asynchrone terugteller met positieve flanktriggering.
Evenals bij de asynchrone voorwaartstellers is de gewenste flanktriggering bepalend voor de te gebruiken flip-flops. Voor positieve flanktriggering zijn Master/ Slave flip-flops nodig. Bij negatieve flanktriggering levert het gebruik van Master/ Slave flip-flops overeenkomstig het gebeuren in figuur 3/6.9-8 verkeerde uitgangscodes.

3. De asynchrone voorwaarts-achterwaarts tellers

Deze teller moet dus zowel op kunnen tellen als af kunnen trekken. Dit wordt bereikt, door in de kloklijn exclusieve-OR poorten op te nemen. Met een digitaal signaal (het niveau op de andere poort ingang) kan men het ingangssignaal inverteren. In figuur 3/6.9-10 zien we dit schematisch weergegeven. Als signaalijn 'V' logisch 1 is zal de teller terugtellen. De funktie wijkt verder niet af van de hiervoor besproken tellers.

4. De synchrone binaire voorwaarts-teller

Wat bij synchrone tellers onmiddellijk opvalt is dat de schakeling er heel anders uitziet. Aangezien alle flip-flops tegelijkertijd worden geklokt, moeten de ingangen van de flip-flops van dusdanige logische niveaus worden voorzien, dat de gewenste funktie wordt bereikt. Hoe de schakeling telt, is afhankelijk van de gebruikte soort flip-flops. Of de triggering op de positieve of de negatieve flanken plaats vindt en het flip-flop principe (bijvoorbeeld Master/ Slave) heeft op het uitgangssignaal geen invloed. Hoogstens is dit bepalend voor het moment waarop de code op de uitgangen verschijnt.

De stuursignalen op de ingangen moeten aan de hand van de gewenste funktie worden berekend. Het volgende voorbeeld laat zien op welke wijze men daartoe te werk dient te gaan.

Opgave is een synchrone modulo-6-voorwaartsteller te ontwerpen, die een binaire uitgangscode levert. Als flip-flops moeten positief getriggerde JK-Master/ Slave flip-flops worden gebruikt.

Aangezien de teller de waarden van 0 (0000_{BCD}) tot 5 ($=0101_{BCD}$) moet kennen zijn drie flip-flops nodig (2^3-1=7). De twee redundante (niet gebruikte) standen zijn $110_{(BCD)}$ - $111_{(BCD)}$.

6.9 Digitale telschakelingen

voor	na	voorbereiding		verklaring
Qn	Qn+1	Jn	Kn	(na afloop van de klokpuls)
0	0	0	d*	logisch 0 houden (bewaren)
0	1	1	d*	uitgang gaat van 0 naar 1 (zetten)
1	0	d*	1	uitgang gaat van 1 naar 0 (terugzetten)
1	1	d*	0	log. 1 houden (bewaren)

*Alle plaatsen waar een d staat kunnen zonder invloed op de werking op dat moment zowel logisch 0 als 1 zijn. De funktie van de JK-flip-flop wordt daardoor niet beïnvloed (d= don't care).

Qn : Q1 uitgang net voor de klokpuls

Qn+1 : Q1 uitgang direkt na de klokpuls

Figuur 3/6.9-11: Waarheidstabel van JK-(Master/Slave) flip-flops.

Nu moeten we even de waarheidstabel van een JK-Master/Slave flip-flop voor de geest halen (figuur 3/6.9-11). Als bijvoorbeeld de uitgang van een flip-flop van logisch 0 naar logisch 1 moet gaan, dan moet de J-ingang logisch 1 zijn. Het niveau op de K-ingang maakt niets uit.

Als volgende stap wordt nu de gewenste uitgangscode in een tabel gezet. De klok geeft aan, wanneer welke uitgangscombinatie moet verschijnen, zie figuur 3/6.9-12. De uitgangscombinaties hebben een oplopende waarde. In principe kan elke gewenste telvolgorde worden gerealiseerd. De $Q1_C$-uitgang is de MSB (most significant bit = bit met de hoogste waarde). Naast de uitgangswaarden zet men de benodigde ingangscondities (voorbereiding op de volgende klokpuls). Bijvoorbeeld wisselt de $Q1_{A\text{-uitgang}}$ bij de eerste klokpuls van logisch 0 naar logisch 1, derhalve moet op de ingang van de A-flipflop voor de conditie $J_A = 1$, $K_A = d$ worden gezorgd. Na de zevende klokpuls begint de telcyclus weer van voren af aan.

Elke gewenste conditie op de diverse ingangen moet met digitale logika worden gerealiseerd. Met behulp van KV-diagrammen is de eenvoudigste schakeling voor elke ingang snel gevonden. De KV-diagrammen van dit voorbeeld vindt u in figuur 3/6.9-13. Experimenteer maar eens met deze schakeling! Ze is eenvoudig genoeg om op een experimenteer boardje te worden gezet. In figuur 3/6.9-14 zijn de afzonderlijke tellerstappen te volgen.

5. De synchrone binaire terugteller

In wezen verschilt de synchrone versie van de terugteller alleen maar in de ge-

6.9 Digitale telschakelingen

funktietabel van de synchrone modulo 6 teller.

	uitgangen			ingangsvoorbereiding					
klok	$Q1_C$	$Q1_B$	$Q1_A$	J_C	K_C	J_B	K_B	J_A	K_A
0	■ 0	■ 0	● 0	0	d	0	d	1	d
1	■ 0	● 0	○ 1	0	d	1	d	d	1
2	■ 0	□ 1	● 0	0	d	d	0	1	d
3	● 0	○ 1	○ 1	1	d	d	1	d	1
4	□ 1	■ 0	● 0	d	0	0	d	1	d
5	○ 1	■ 0	○ 1	d	1	0	d	d	1
Reset	0	0	0						
6	1	1	0	redundant					
7	1	1	1						

deze wisseling treedt op bij de volgende flank van de klok.
De voorbereiding van de J en K ingangen gebeurt zoals rechts staat aangegeven

■ logisch 0 houden (bewaren)

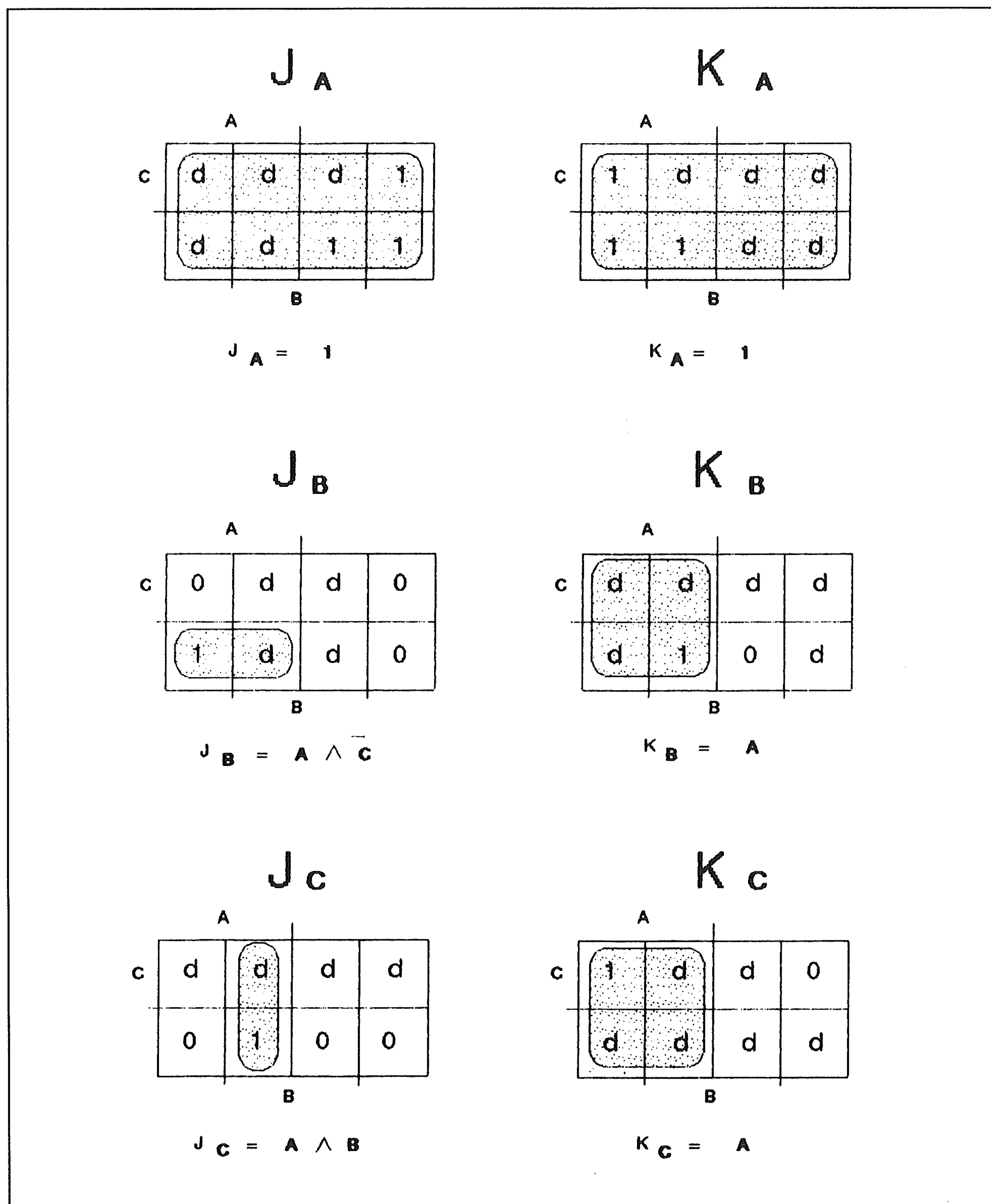
● uitgang gaat van logisch 0 naar logisch 1 (zetten)

□ logisch 0 houden (bewaren)

○ uitgang gaat van logisch 1 naar logisch 0 (terugzetten)

Figuur 3/6.9-12: Funktietabel van een modulo-6-teller.

6.9 Digitale telschakelingen



Figuur 3/6.9-13: KV-diagrammen van de modulo-6-teller.

wenste uitgangscode van de voorwaarts-teller.
Aangezien bij synchrone tellers de uitgangscode afhankelijk is van de voorbereiding van de ingangen, met andere woorden de schakelingen aan de ingangen, gaat het realiseren van een terugteller hetzelfde als van een voorwaarts-teller, maar met andere logika op de ingangen. Op synchrone voorwaarts- en achterwaartstellers gaan we hier niet verder in.

Set en Reset mogelijkheden

Als men de hierboven beschreven teller nabouwt en men zet de voedingsspanning aan, dan is niet te voorspellen welke waarde de uitgangen zullen hebben. Om dus te zorgen dat de teller met zekerheid bij $0000_{(BCD)}$ begint met tellen moet de teller op één of andere wijze op deze waarde worden gezet. Dit op 0 zetten wordt in het Engels 'reset' genoemd, in tegenstelling met het op 1 zetten dat 'set' wordt genoemd. Als van een telschakeling niet alle ingangen op 0 of 1 worden gezet, maar de teller als zodanig op een bepaalde waarde wordt gezet, spreekt men van 'preset'. Voor het op nul zetten hebben de meeste flip-flops een R (reset) ingang.

Resetten is ook noodzakelijk voor tellers, die niet alle mogelijke waardes mogen doorlopen. Denk bijvoorbeeld aan een asynchrone decadeteller (=10-teller).
De waarden 0 (0000_{BCD}) tot 9 (1001_{BCD}) mogen voorkomen, alle waarden daarboven echter niet.

Het resetten kan op twee verschillende manieren worden bereikt. De nog komende klokpuls zorgt via de nodige logika voor het resetten van de flip-flops. Dit is relatief ingewikkeld.

De meer gebruikelijke methode is, om de eerste 'verboden' uitgangscombinatie te decoderen en hieruit een resetpuls te genereren. In figuur 3/6.9-15 is deze methode in praktijk gebracht. Het resetten met deze methode heeft een nadeel. De niet gedefinieerde (verboden) uitgangscombinatie komt voor, zij het dan slechts zeer kort. In de praktijk blijkt dit echter geen al te groot bezwaar, tenzij een achter de teller liggend geheugen de waarde bewaart. In de meeste gevallen kan een simpele enable schakeling in dat geval dit probleem ondervangen.

Als een teller op een bepaalde waarde moet worden ingesteld voordat de telcyclus gaat lopen (preset), dan kunnen de individuele Set-ingangen van de afzondelijke flip-flops worden gebruikt. Iedere flip-flop, die een logische 1 op de uitgang moet leveren, wordt geset.

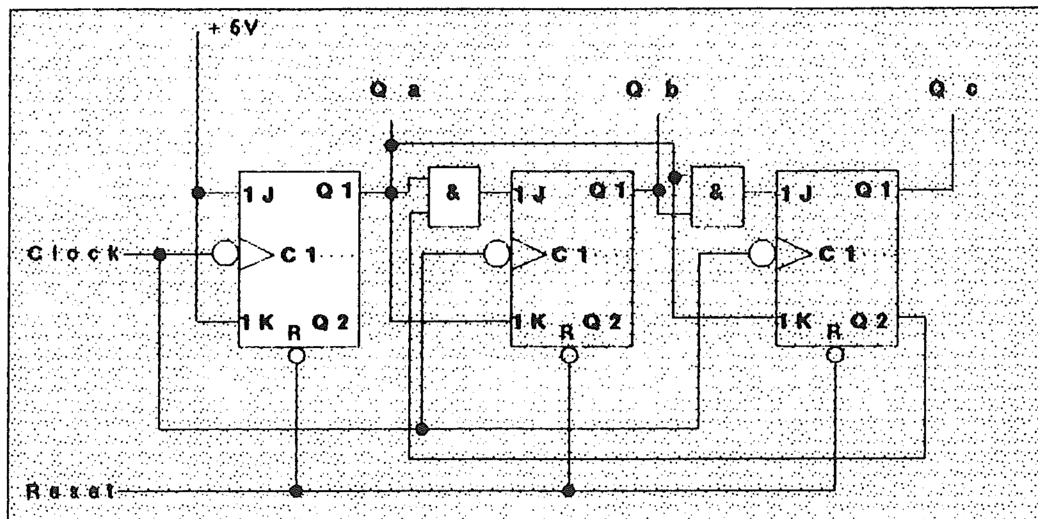
De Set- en Reset-ingangen van flip-flops kunnen zowel synchroon als asynchroon werken. Als deze ingangen asynchroon werken, dan zal de uitgang van de flipflop onmiddellijk na het verschijnen van het Set- of Reset-signaal de betreffende uitgangstoestand aannemen. Bij synchrone werking wordt het Set- of Reset-signaal aangelegd, maar de eigenlijke werking komt pas na de volgende klokpuls. Het Set- of Reset-signaal moet gedurende de klokpuls stabiel blijven. Het synchroon setten en resetten van tellers zorgt ervoor, dat onder alle omstandigheden het uitgangssignaal synchroon is met de klok.

In ieder geval moet worden voorkomen, dat de Set- en Reset-ingangen gelijktijdig actief zijn. Het resulterende uitgangsniveau is dan niet te voorspellen. In figuur 3/6.9-16 zien we een praktische schake-

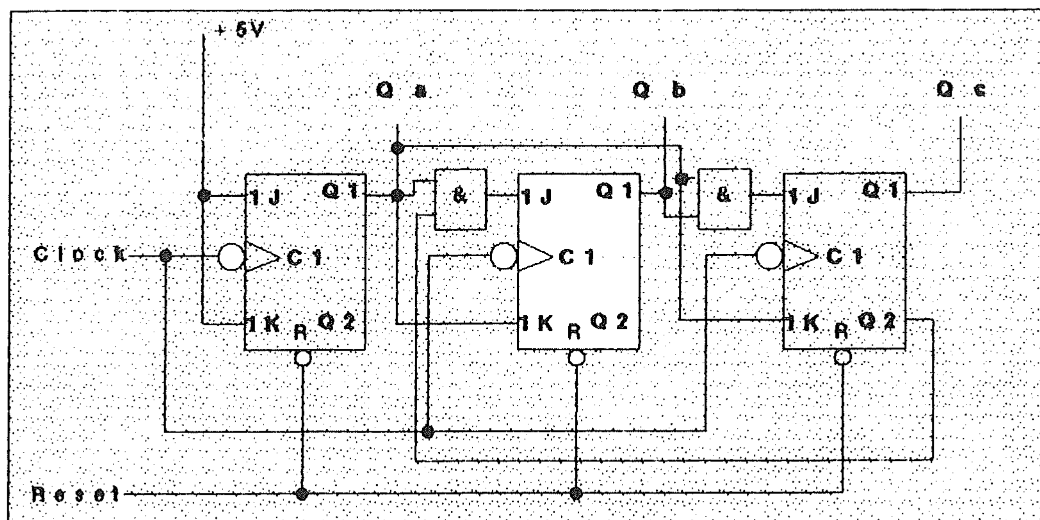
6.9 Digitale telschakelingen

I. De teller wordt gereset: een negatieve puls op de Reset-ingang zorgt voor een gedefinieerde uitgangstoestand voor alle flip-flops. Alle Q1-uitgangen zijn logisch 0 en alle Q2-uitgangen zijn logisch 1.

II. Na de eerste klokpuls klapt de eerste flip-flop om. Daardoor wordt Qa logisch 1. De uitgangen leveren tellerstand 1.



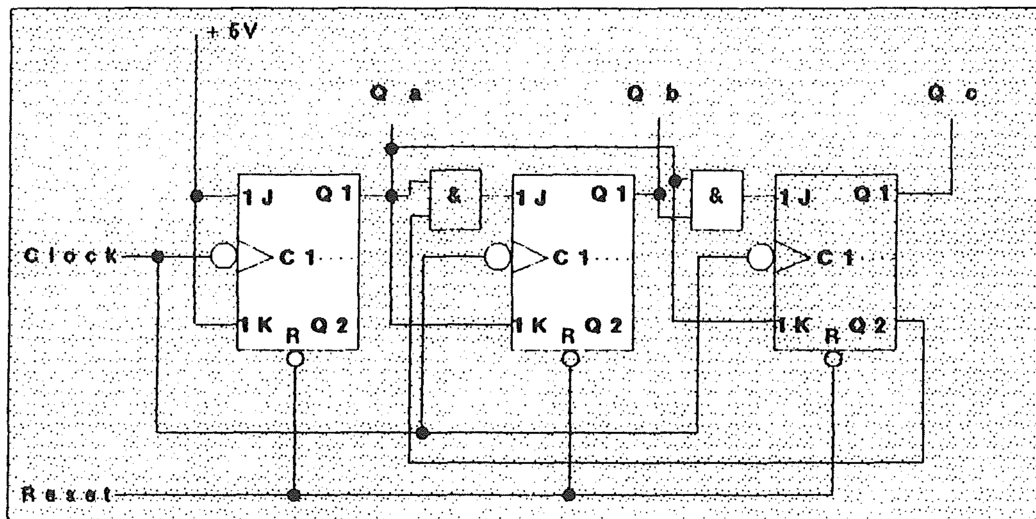
III. De tweede negatieve flank van de klok set de tweede flip-flop en reset de eerste. Daarmee is Qa weer logisch 0 en Qb logisch 1. De tellerstand is nu 2.



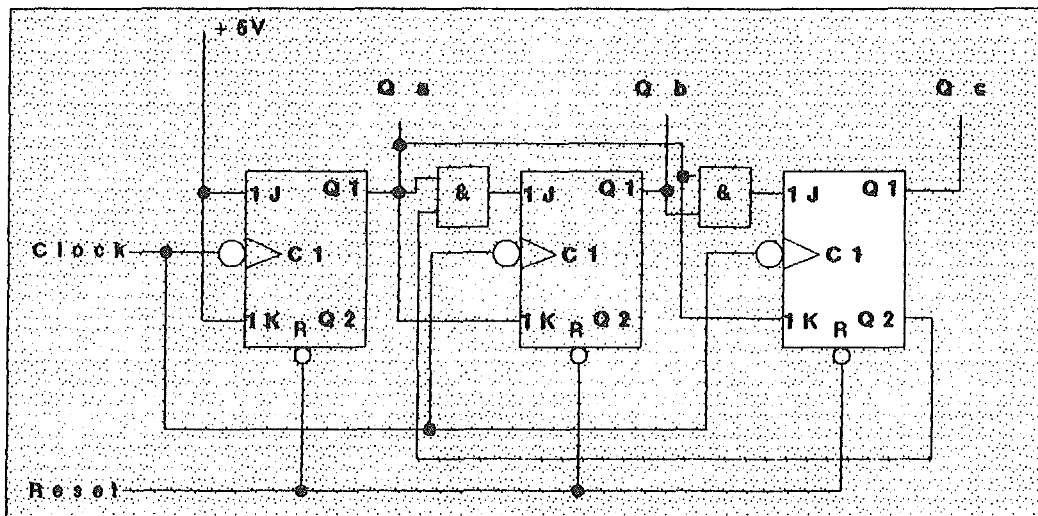
Figuur 3/6.9-14 (deel 1): Het verloop van de telcyclus van een synchrone modulo-6-teller.

6.9 Digitale telschakelingen

IV. De derde negatieve flank van de klok zorgt voor het omklappen van de eerste flip-flop. Qa wordt dus logisch 1.
De tweede flip-flop bewaart zijn toestand, dus Qb blijft logisch 1.



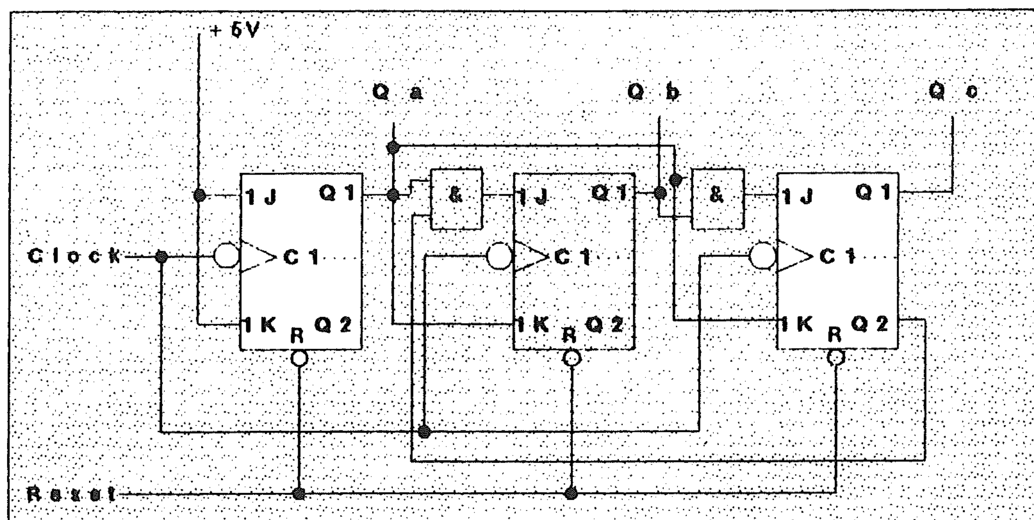
V. De vierde negatieve flank van de klok set de tweede flip-flop en reset de eerste.
Ook de derde wordt geset. De tellerstand is nu 4.



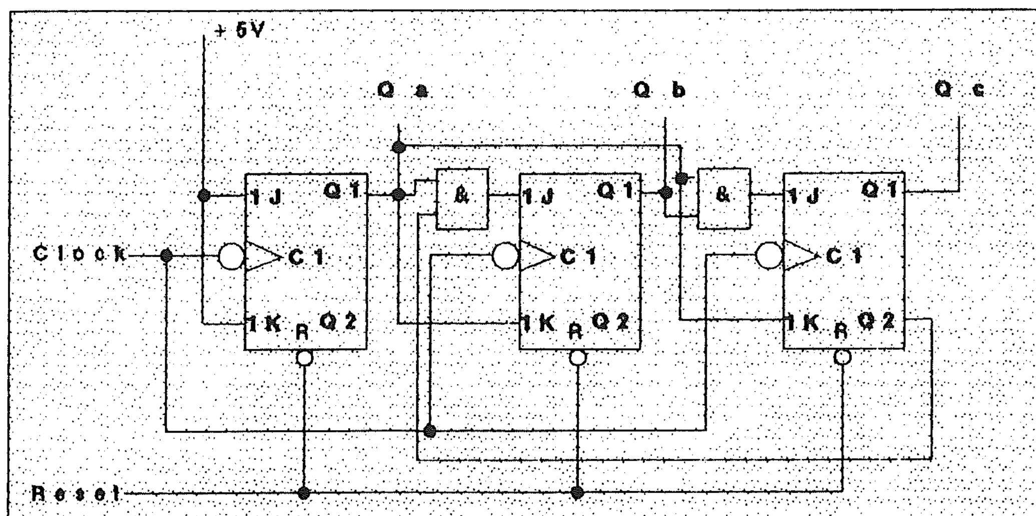
Figuur 3/6.9-14 (deel 2): Het verloop van de telcyclus van een synchrone modulo-6-teller.

6.9 Digitale telschakelingen

VI. Op de vijfde negatieve flank van de klok klapt de eerste flip-flop weer om. Qa wordt logisch 1.
De tweede flip-flop blijft gereset en de derde flip-flop bewaart zijn waarde, Qc blijft logisch 1.

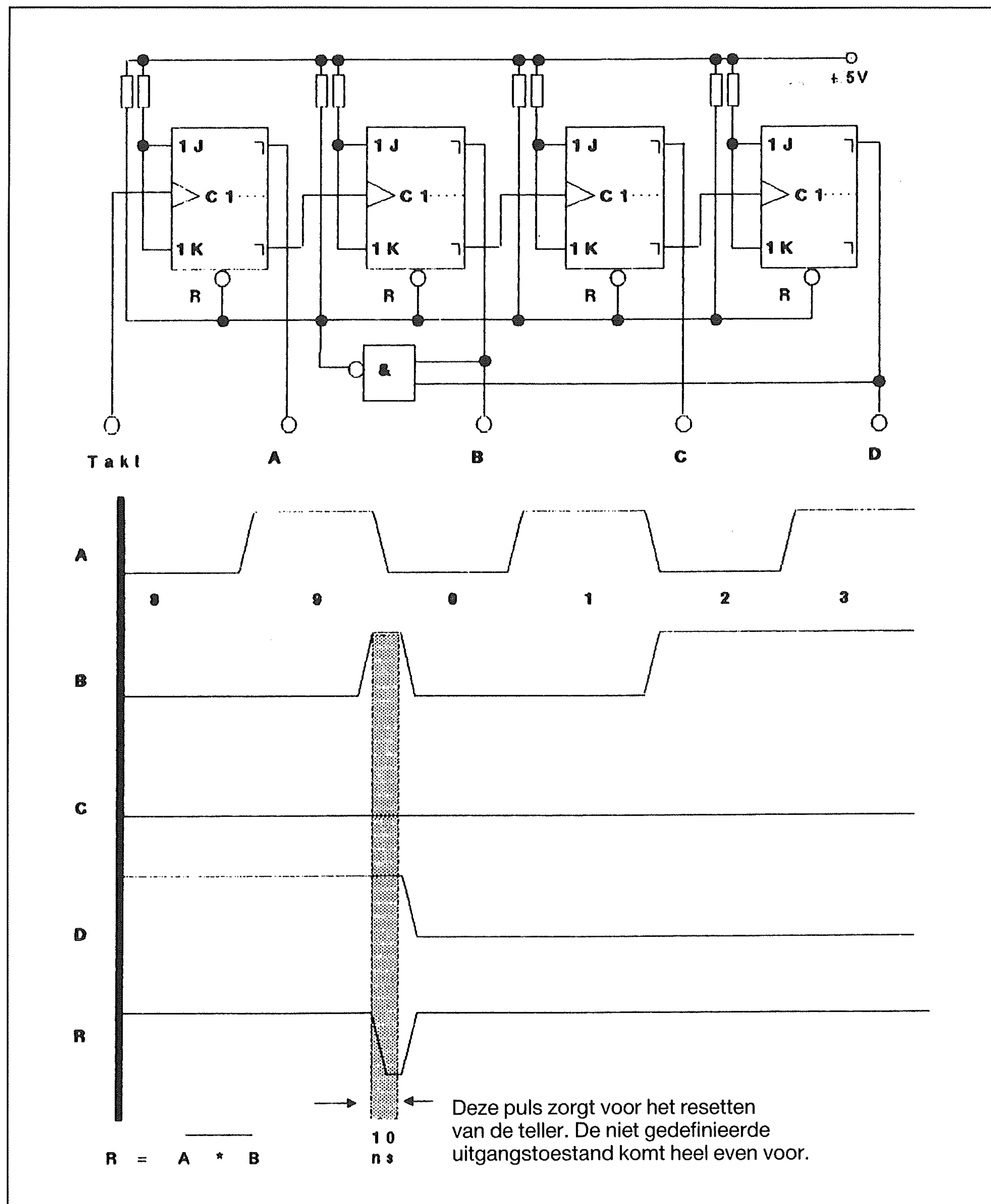


VII. De zesde negatieve flank van de klok zet alle flip-flops weer op 0 terug en de telcyclus herhaalt zich.



Figuur 3/6.9-14 (deel 3): Het verloop van de telcyclus van een synchrone modulo-6-teller.

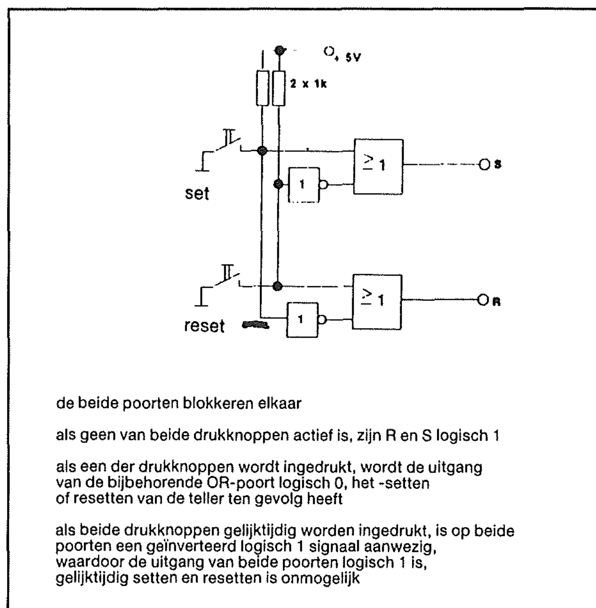
6.9 Digitale telschakelingen



Figuur 3/6.9-15: Het maken van een resetpuls.

6.9 Digitale telschakelingen

ling die deze toestand voorkomt, door de Set- en Reset-signalen elkaar te laten blokkeren. Ook is het aan te raden en bij CMOS zelfs noodzakelijk om de Set- en Reset-ingangen van een pull-up of pull-down, afhankelijk van de gebruikte logika, weerstand te voorzien om storingen te voorkomen.



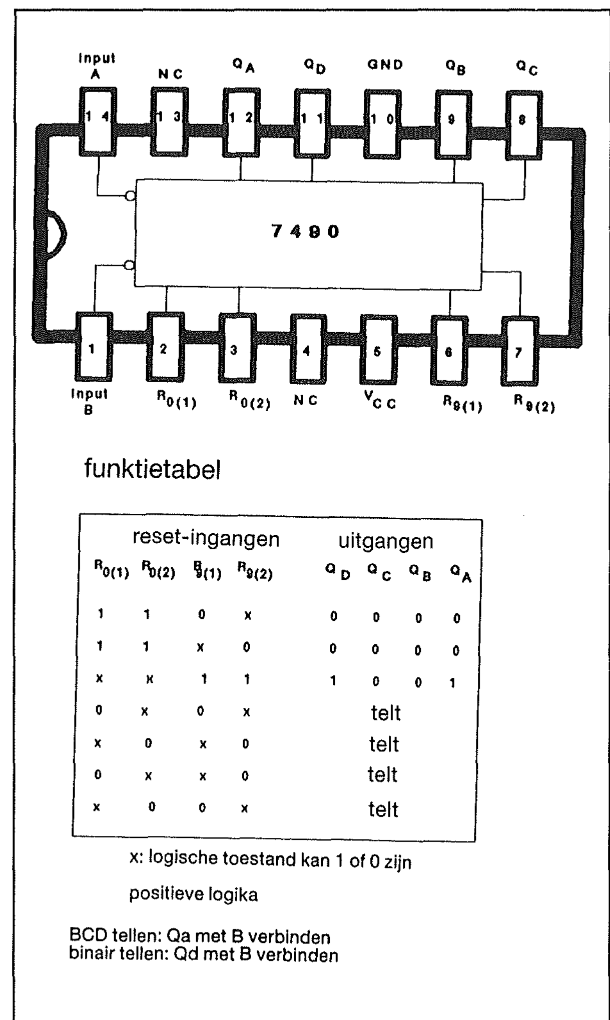
Figuur 3/6.9-16: Onderlinge blokkering van Set en Reset.

Sommige flip-flops hebben een zogenaamde voorkeurstoestand. Intern in de flip-flops zijn opzettelijk kleine verschillen in de diverse transistoren aangebracht, of heeft men in de collectorkring van de transistoren weerstanden aangebracht, zodat de flip-flop bij het inschakelen van de voeding een bepaalde waarde aanneemt. Deze flip-flops zorgen voor een bekende uitgangstoestand bij inschakelen van de teller waarin zij worden toegepast.

De oplossing in geïntegreerde vorm

Voor de ingewikkeld tellers is de industrie ons gelukkig te hulp gekomen. Een zeer populair teller-IC is de in figuur 3/6.9-17 weergegeven SN 7490. De SN 7490 kan als decade-teller worden geschakeld, als ook als binaire (modulo-16) teller. De werking is eenvoudig uit de funktietabel op te maken.

Een lid van de familie van synchrone voorwaartstellers is de SN 74162 van figuur 3/6.9-18. Via de ingangen Data-A tot en met Data-D is de teller te presetten. Overname van de op de Data-ingangen aangeboden waarde gebeurt synchroon met de klok.



reset-ingangen				uitgangen			
$R_{0(1)}$	$R_{0(2)}$	$R_{9(1)}$	$R_{9(2)}$	Q_D	Q_C	Q_B	Q_A
1	1	0	x	0	0	0	0
1	1	x	0	0	0	0	0
x	x	1	1	1	0	0	1
0	x	0	x	telt			
x	0	x	0	telt			
0	x	x	0	telt			
x	0	0	x	telt			

x: logische toestand kan 1 of 0 zijn

positieve logika

BCD tellen: Qa met B verbinden
binair tellen: Qd met B verbinden

Figuur 3/6.9-17: Het teller-IC 7490.

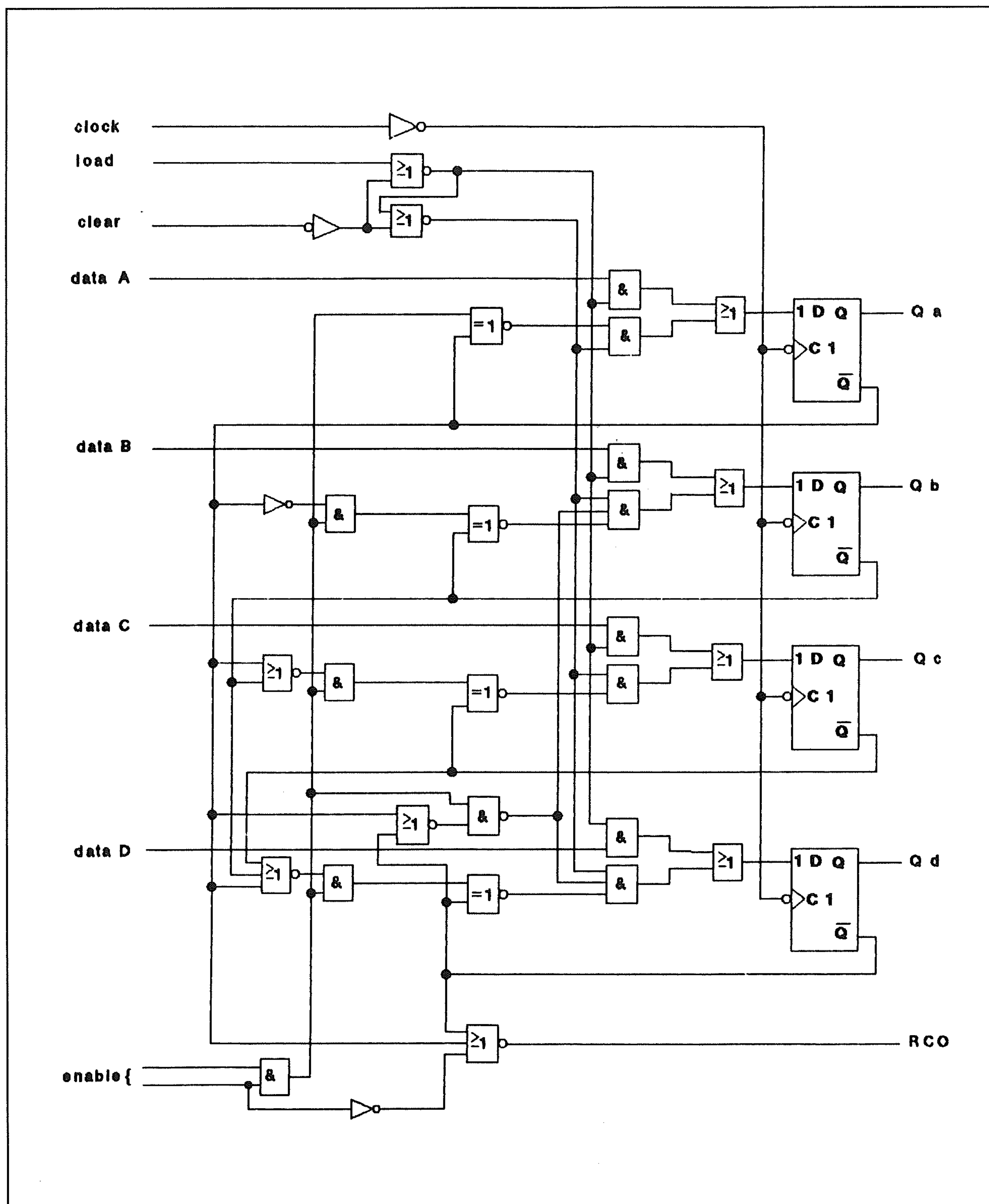
6.9 Digitale telschakelingen

Experimenteren

Zoals vaak levert experimenteren net dat beetje extra inzicht, dat de materie doorzichtig maakt. Het nabouwen van de schakelingen op een experimenteerbord zal niet veel problemen geven. De gegevens van de gebruikte IC's kunt u in de datatabellen van deel 6 wel terugvinden. Een probleempje doet zich echter voor. We hebben te maken met geklokte logika, dus moeten we een kloksignaal hebben. Een pulsgenerator kan hier uitkomst bieden. Natuurlijk kan het tellen ook wel met een drukknop worden bestuurd, echter drukknoppen hebben nogal eens de neiging te denderen, waardoor de schakeling niet 1 maar ..tig klokpulsen ziet en ongedefinieerd lijkt te tellen. De drukknop zal dus noodzakelijkerwijs door de schakeling moeten worden gevolgd die de kontaktdender onderdrukt. Zie bijvoorbeeld de monostabiele multivibrator van figuur 3/6.9-19c!

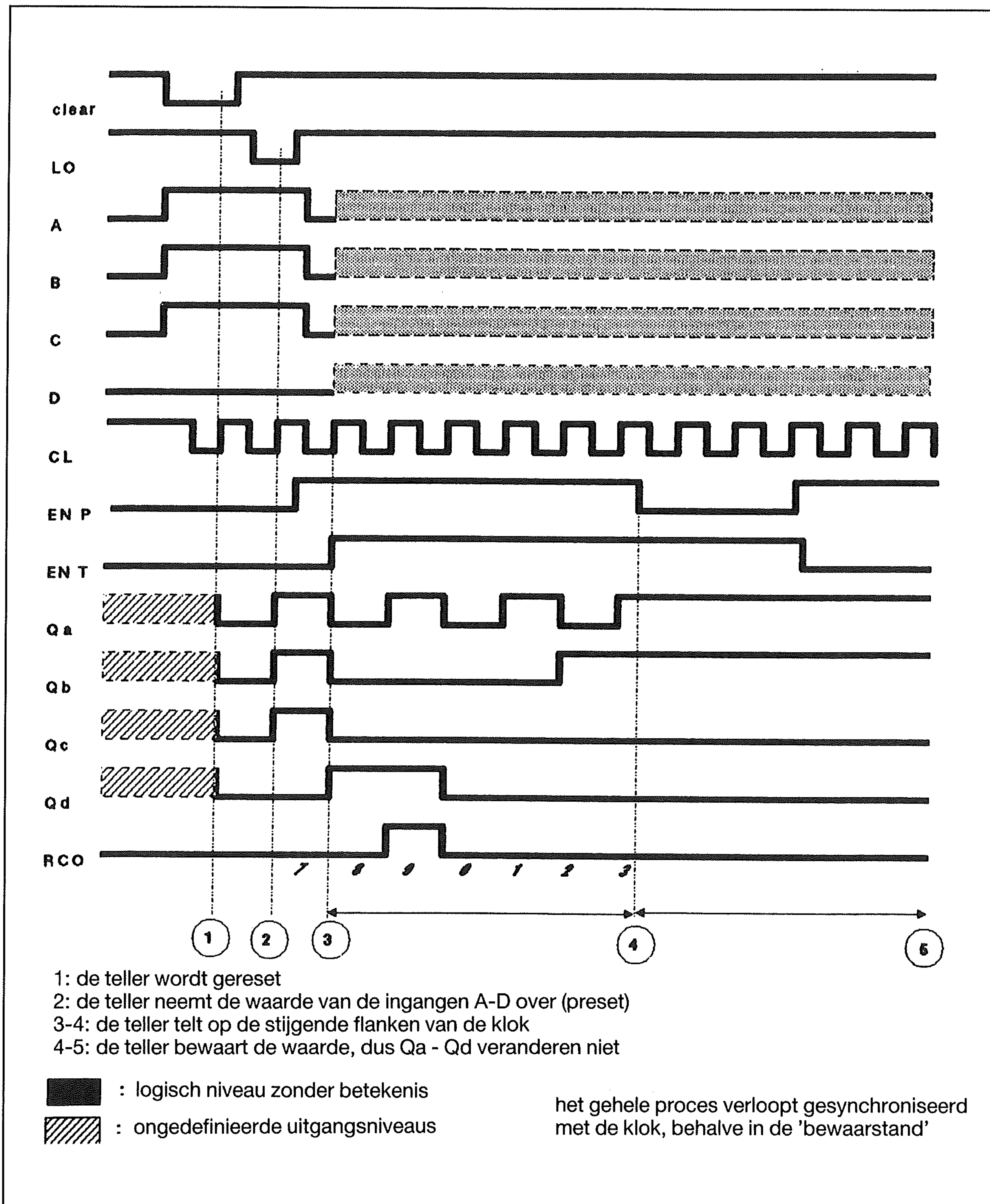
Na verloop van tijd wordt het drukken op de knop toch een vervelende aangelegenheid. Om u het leven te veraangenamen is in figuur 3/6.9-19a een eenvoudige klokschakeling opgenomen, opgebouwd rond de astabiele multivibrator ICM 7242. Met de RC-constante kan een groot frequentiebereik worden ingesteld. Als voor de weerstand een potentiometer wordt genomen, kan de frequentie traploos worden geregeld.

6.9 Digitale telschakelingen



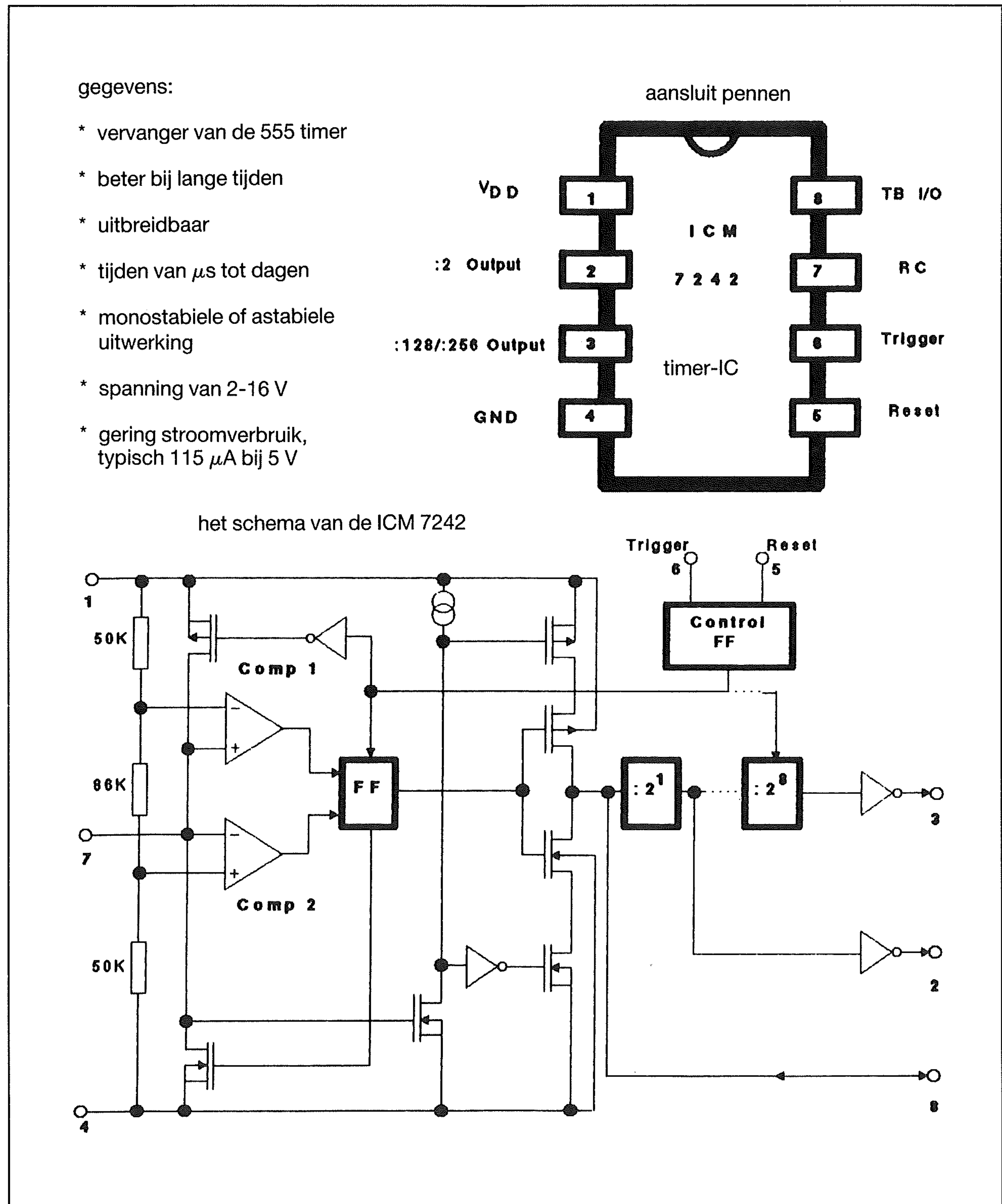
Figuur 3/6.9-18a: Het 'interieur' van de synchrone decade-teller 74S162.

6.9 Digitale telschakelingen



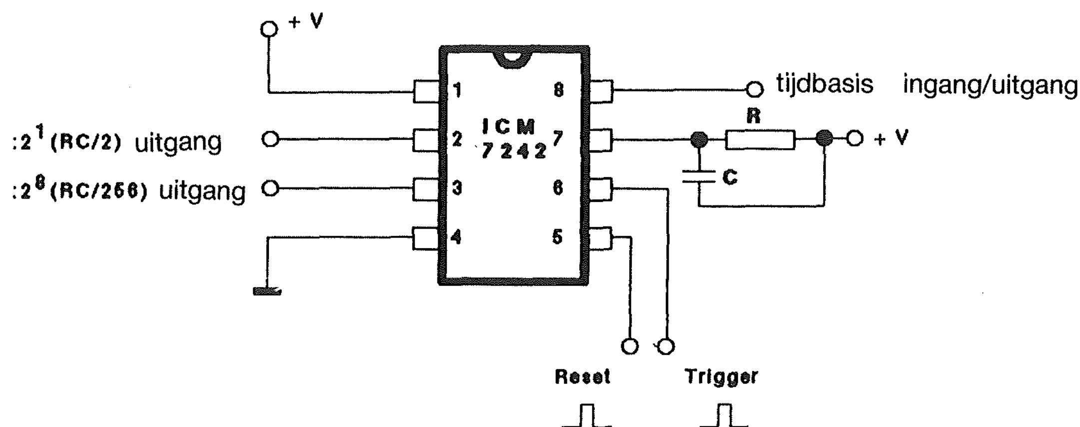
Figuur 3/6.9-18b: Funktietabel van de decade-teller 74S162.

6.9 Digitale telschakelingen



Figuur 3/6.9-19a: Timer IC voor lange tijden ICM 7242.

6.9 Digitale telschakelingen



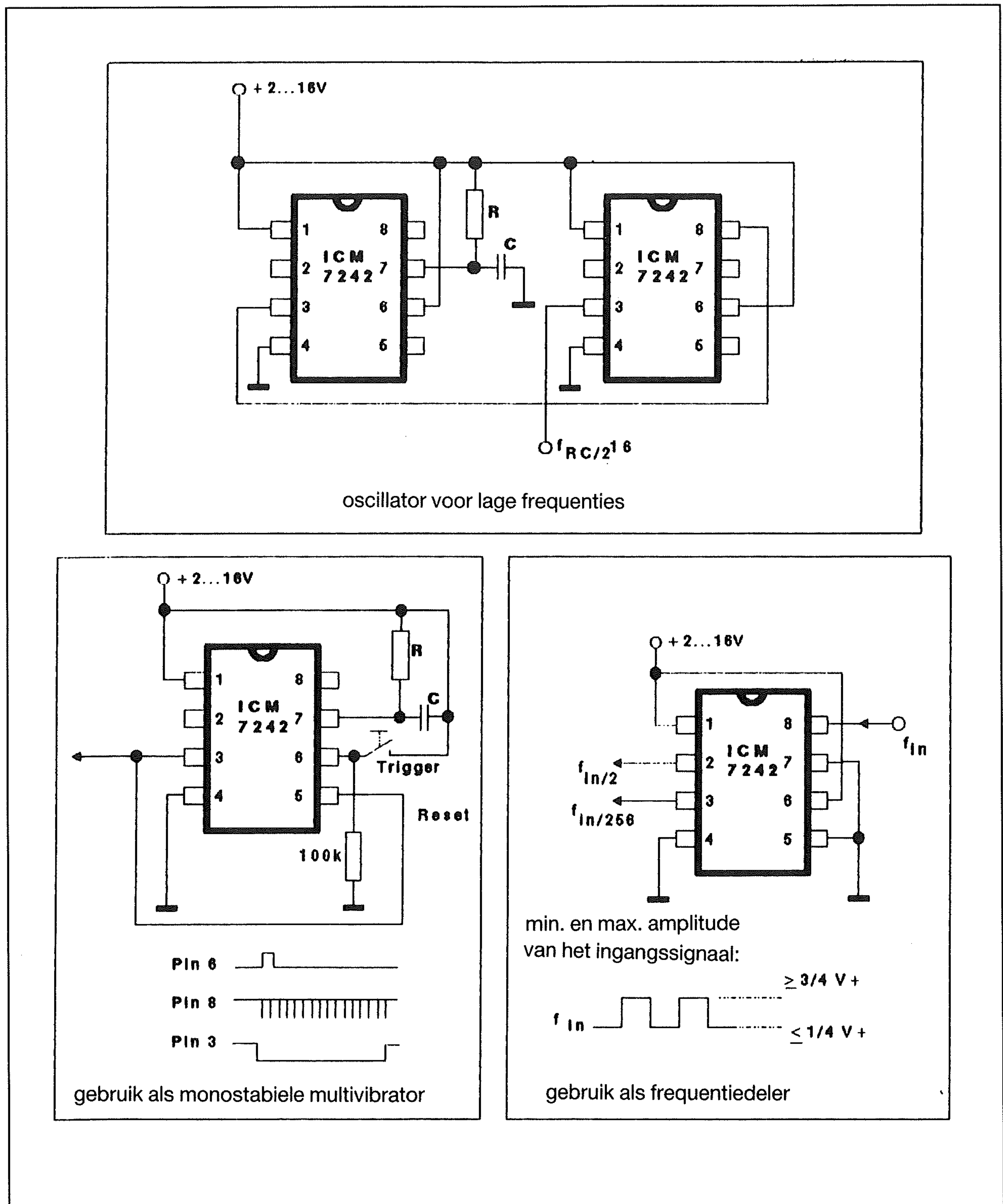
opmerking: de deleruitgangen 2_1 en 2_8 zijn geïnverteerd;
intern zijn ze reeds van pull-up weerstanden voorzien

maximale specificaties

voeding (+V naar GND)	max. 18 V
ingangsspanningen (pennen 5, 6, 7 en 8)	GND – 0.3 V . . +V + 0.3 V
Ausgangsdauerstrom	max. 50 mA
dissipatie	200 mW
werktemperatuur	-25°C tot 85°C
soldeertemperatuur (10 sec.)	300°C
aanbevolen waarden RC componenten:	
weerstand R	van 1kΩ tot 20 MΩ
condensator C	van 1nF tot 10 mF
berekening van de uitgangsfrequentie **(PIN 8)**	1,0 x R x C
tijdafwijking	max. 5 %

Figuur 3/6.9-19b: Astabiele Multivibrator met ICM 7242.

6.9 Digitale telschakelingen



Figuur 3/6.9-19c: Schakelvoorbeelden met de ICM 7242.

6.9 Digitale telschakelingen

3/6.12

Samenstelling en werking logika-families

Inhoud

3/6.12.1 Inleiding en overzicht
(verschenen in de 19e aanvulling)

3/6.12.2 Samenstelling en werking van de TTL logika
(verschenen in de 19e aanvulling)

3/6.12.3 Samenstelling en werking van de CMOS logika
(verschenen in de 20e aanvulling)

3/6.12.4 Samenstelling en werking van de ECL logika
(verschenen in de 26e aanvulling)

3/6.12.5 Samenstelling en werking van de DTL logika
(verschenen in de 48e aanvulling)

3/6.12.6 Samenstelling en werking van de BI(C)MOS
(verschenen in de 48e aanvulling)

[1] Dit hoofdstuk heeft een eigen inhoudsopgave

3/6.12.4

Samenstelling en werking van de ECL logika

Inleiding
Alle andere transistor logika families, zoals RTL, DTL en TTL, lijden aan een algemene en fundamentele beperking wat betreft de schakelsnelheid. Deze beperking onstaat doordat transistoren in verzadiging gestuurd worden, met als gevolg een hoge propagation delay.

Bij ECL logika worden geen transistoren in verzadiging gestuurd, maar wordt ervoor gezorgd dat de transistoren maximaal schakelen van een gesperde toestand naar een punt in het actieve gebied.
Met actief gebied wordt bedoeld een punt waarin een verhoging van de basisstroom ook een verhoging van de collectorstroom tot gevolg heeft.
De propagation delay van ECL logika zal daardoor aanzienlijk lager zijn dan bijvoorbeeld van de TTL logika.
Een MECL III poort heeft een propagation delay van ongeveer 1 ns.

Overige voordelen van ECL
Andere voordelen van ECL logika zijn:
- lage uitgangsimpedantie;
- hoge fan-out;
- lage eigen storingsbijdrage.

Toepassingsvoorbeelden
Door de lage propagation delay en daardoor hoge maximum schakelsnelheid wordt ECL vaak toegepast wanneer snelheid belangrijk is.

Voorbeelden hiervan zijn:
- radar-systemen;
- medische elektronica;
- data-transmissie;
- computers;
- snelle A/D converters.

Algemeen schema
Figuur 3/6.12.4-1 geeft het schema van een MECL II (N)OR-poort.

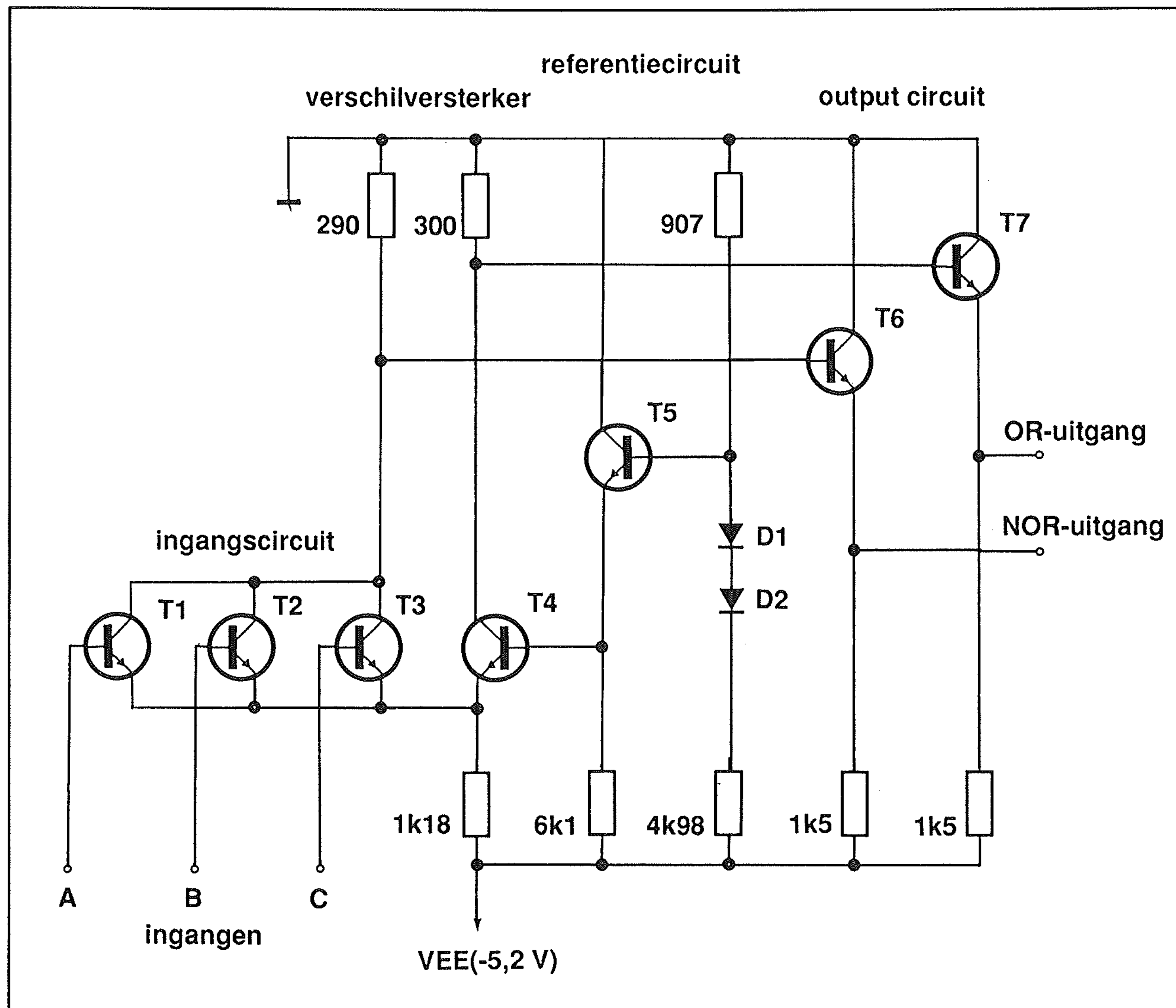
In dit schema zijn vier verschillende delen te onderscheiden, namelijk:
- het ingangs circuit;
- de verschil versterker;
- het referentie circuit;
- het output circuit.

De verschil versterker
Het feitelijke schakelwerk gebeurt in de verschil versterker. Het schema van deze schakeling is nog eens getekend in figuur 3/6.12.4-2.

Uit deze figuur volgt dat beide emitters met elkaar verbonden zijn, hetgeen karakteristiek is voor ECL logika en waaraan de naam van de familie wordt ontleend.

Emittor Coupled Logic staat immers voor emitter gekoppelde logika!

6.12 Samenstelling en werking logika-families



Figuur 3/6.12.4-1: Het schema van een MECL II (N)OR-poort.

De ingangsspanning U_{in} staat op de basis van T3 en een vaste referentiespanning U_{ref} op de basis van T4.
Is de ingangsspanning voldoende lager dan de referentiespanning, dan zal transistor T3 sperren en zal er door T4 een stroom gaan lopen. De weerstanden Rc4, Re en de referentiespanning zijn echter zo gekozen dat transistor T4 niet kan verzadigen.

Het stijgen van de ingangsspanning heeft tot gevolg dat de stroom door T3 zal toenemen en de stroom door T4 zal afnemen.
De stromen door T3 en T4 zijn gelijk zodra de ingangsspanning even hoog is als de referentiespanning.

Stijgt de ingangsspanning verder, dan zal de emmitterspanning U_e ook gaan stijgen ($U_e = U_{in} - U_{be}$, U_{be} is konstant), waardoor T4 uit geleiding zal gaan. Rc3 is zo gekozen dat T3 niet kan verzadigen.
T3 en T4 werken dus in het actieve gebied en verzadigen niet.

6.12 Samenstelling en werking logika-families

Samenvattend komt het er op neer dat een verandering van de ingangsspanning ervoor zorgt dat de stroom door de verschil versterker als het ware omschakelt tussen de ene en de andere transistor.
Een voordeel van deze schakeling is dat de stroomopname, ongeacht de uitgangstoestand, vrijwel konstant is. Dit betekent echter dat de schakeling veel dissipeert. Een (N)OR poort volgens figuur 6/3.12.4-1 dissipeert ongeveer 30 mW.

De overdrachtskarakteristiek
In figuur 6/3.12.4-3 is de berekende overdrachtskarakteristiek getekend van de OR-uitgang van de poort uit figuur 6/3.12.4-4.

Stel dat de ingangsspanning U_{in} minder negatief is dan de referentiespanning. Dit betekent dat transistor T4 is gesperd en er geen stroom door Rc4 zal lopen. De uitgangsspanning is dan gelijk aan:

$$U_u = U_{rc4} - U_{be(T7)}$$

De spanning U_{rc4} wordt gegeven door stroom maal weerstand, dus:

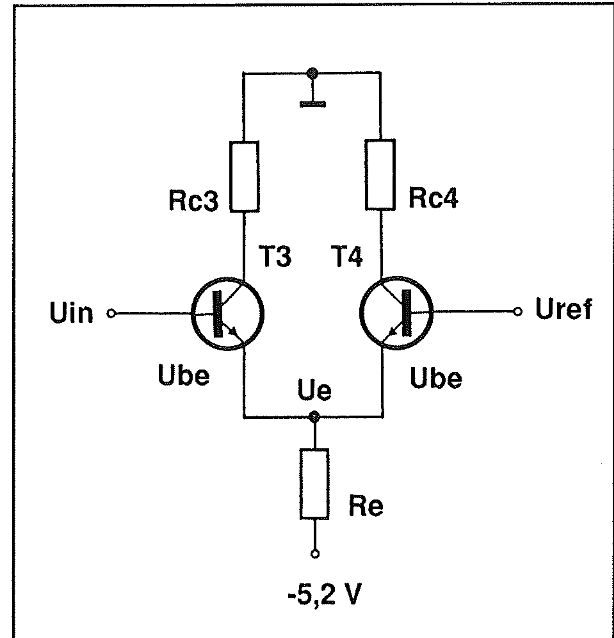
$$U_{rc4} = -I_{rc4} * R_{c4}$$

Aangezien er geen stroom door Rc4 loopt, basisstromen worden verondersteld verwaarloosbaar klein te zijn, is de spanning U_{rc4} gelijk aan "0 V".

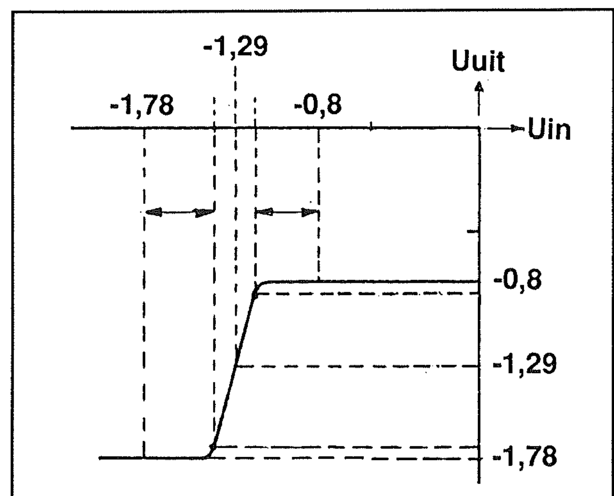
De basis-emitter spanning U_{be} van een geleidende transistor is ongeveer 0,8 V. Het resultaat is een uitgangsspanning van:

$$U_u = 0 - 0,8\ V = -0,8\ V$$

Een spanning van -0,8 V wordt gedefinieerd als logisch 1.



Figuur 3/6.12.4-2: Het schema van de verschil versterker, die verantwoordelijk is voor het schakelgedrag.



Figuur 3/6.12.4-3: De berekende overdrachtskarakteristiek van de schakeling.

Zodra de ingangsspanning negatiever wordt dan de referentiespanning, zal T3 sperren en T4 geleiden. De uitgangsspanning is nu als volgt te berekenen.

Op de emitter van T4 staat:

$$U_e = U_{ref} - U_{be}$$
$$U_e = -1{,}29\ V - 0{,}8\ V = -2{,}09\ V$$

De stroom door de weerstand Re is:

$$I_e = (U_e - V_{ee}) / R_e$$
$$I_e = (-2{,}09\ V - (-5{,}2\ V)) / 779\ \Omega = 4{,}0\ mA$$

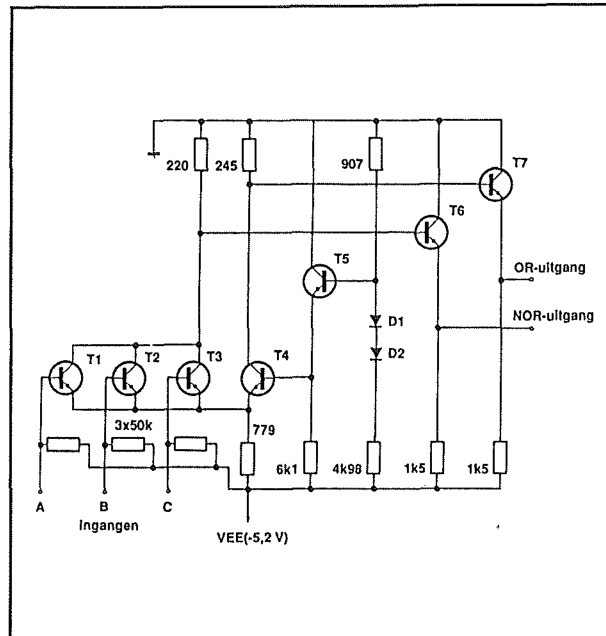
Als de basisstroom verwaarloosd mag worden is de emitterstroom I_e gelijk aan de collectorstroom I_c van T4. De uitgangsspanning wordt nu:

$$U_u = U_{rc4} - U_{be}$$
$$U_u = (-I_{rc4} * R_{c4}) - U_{be}$$
$$U_u = -0{,}98\ V - 0{,}8\ V = -1{,}78\ V$$

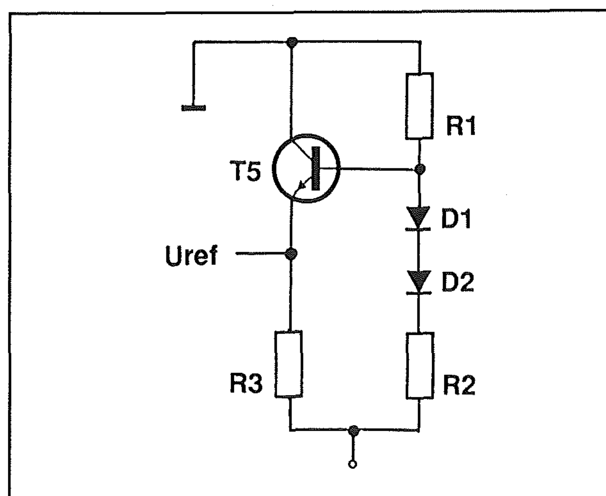
Deze -1,78 V wordt gedefinieerd als logisch 0.

De referentiespanning van -1,29 V blijkt niet meer een willekeurige waarde te zijn, maar ligt precies tussen de beide uitgangsniveau's -0,8 V en -1,78 V in.
Zodra de ingangsspanning groter wordt dan de referentiespanning, verandert de uitgangsspanning van -1,78 V (logisch 0) naar -0,8 V (logisch 1). Deze verandering, van logisch 0 naar 1 of omgekeerd, gebeurt op het moment dat de ingangsspanning net boven of net beneden de referentiespanning komt. Het blijkt dat als de ingangsspanning 75 mV hoger of 75 mV lager is dan de referentiespanning, de uitgangsspanning zijn eindwaarde heeft bereikt (logisch 0 of 1). De referentiespanning bepaalt dus het moment van schakelen.

De referentiespanning
Figuur 6/3.12.4-5 geeft het schema van een referentiecircuit.



Figuur 3/6.12.4-4: Een MECL 10.000 (N)OR poort.



Figuur 3/6.12.4-5: Het referentie circuit.

De spanningsdeler R1/R2 zorgt voor de juiste spanning op de basis van T5. Transistor T5 is als emittervolger geschakeld en zorgt daardoor voor een lage uitgangsweerstand van het referentiecircuit. De dioden D1 en D2 zijn voor de temperatuurcompensatie zodat de refe-

rentiespanning precies tussen de beide uitgangsniveau's blijft liggen.

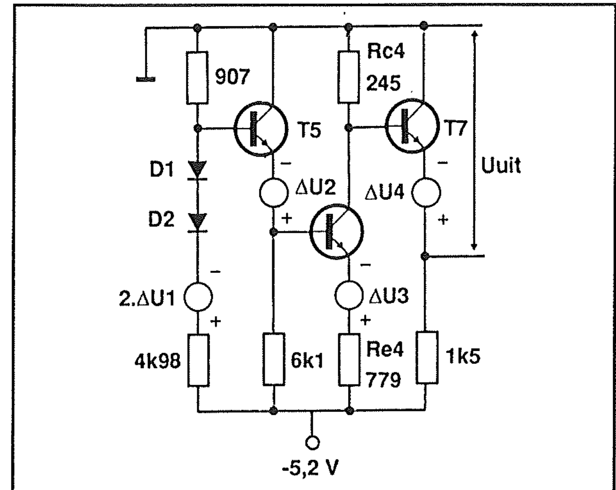
Stijgt de temperatuur, dan daalt de diode of basis-emitter spanning. Deze spanningsverandering van U_{be} zal worden aangeduid met ΔU.

In figuur 3/6.12.4-6 is het vervangingsschema weergegeven van een ECL poort waarin alle spanningsvariaties als gevolg van het temperatuursverloop, met ΔU spanningsbronnen zijn weergegeven.

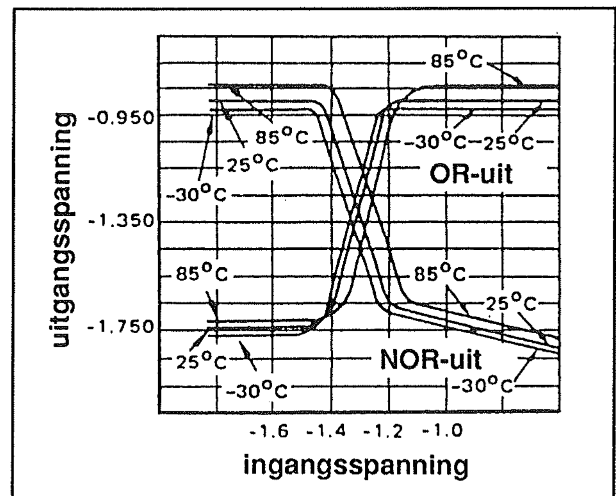
Een temperatuursverhoging heeft tot gevolg dat ΔU groter wordt. De uitgangsspanning van beide logische niveau's zal daardoor positiever worden.

In figuur 3/6.12.4-7 zijn een aantal overdrachtskarakteristieken getekend van een MECL 10.000 poort, als functie van de temperatuur.

Het logisch 1 niveau wordt, zie figuur 3/6.12.4-6, ΔU_4 positiever, omdat de uitgangsspanning volledig bepaald wordt door de basis-emitter spanning van T7 in figuur 3/6.12.4-6.
Logisch 0 niveau verandert minder dan ΔU_4 omdat niet alleen de basis-emitter spanning van T7 maar ook van T4 en T5 daalt.
Blijft de spanning op de basis van T5 gelijk, dan zal de spanning over de weerstand Re4 ($\Delta U_2 + \Delta U_3$) groter worden. De stroom door Re4 en dus ook door Rc4 zal toenemen. Een grotere stroom door Rc4 betekent een grotere spanning over Rc4 met als gevolg dat de uitgangsspanning iets lager zal worden. De uitgangsspanning wordt nu niet ΔU_4 hoger maar iets minder vanwege de grotere spanning over Rc4.



Figuur 3/6.12.4-6: Effect van de temperatuurvariatie.



Figuur 3/6.12.4-7: Overdrachtkarakteristiek als functie van de temperatuur.

De referentiespanning (basisspanning van T4) wordt ΔU_2 hoger, dioden D1 en D2 worden even weggelaten.
Zou de uitgangsspanning bij zowel logisch 0 als 1 ΔU zijn gestegen, dan zou er verder niets gedaan hoeven worden omdat de referentiespanning ook ΔU is gestegen en dus weer precies midden tussen de beide logische niveau's in zou

6.12 Samenstelling en werking logika-families

liggen. Logisch 0 verandert echter iets minder dan ΔU waardoor de referentiespanning ook iets minder dan ΔU moet veranderen om midden tussen de niveau's te blijven.
Deze compensatie van de referentiespanning wordt gedaan door de dioden D1 en D2.
De diodespanning verandert met ΔU_1 waardoor de spanning op de basis van T5 iets lager wordt. De referentiespanning zal iets minder dan ΔU hoger worden, waardoor het schakelpunt in het midden blijft.

Storingsongevoeligheid
Door het geringe ingangsspanningsverschil tussen logisch 0 en 1 is de ECL poort erg gevoelig voor storingen.

Een ECL poort ziet een logisch 1 op de ingang als de ingangsspanning hoger is dan:

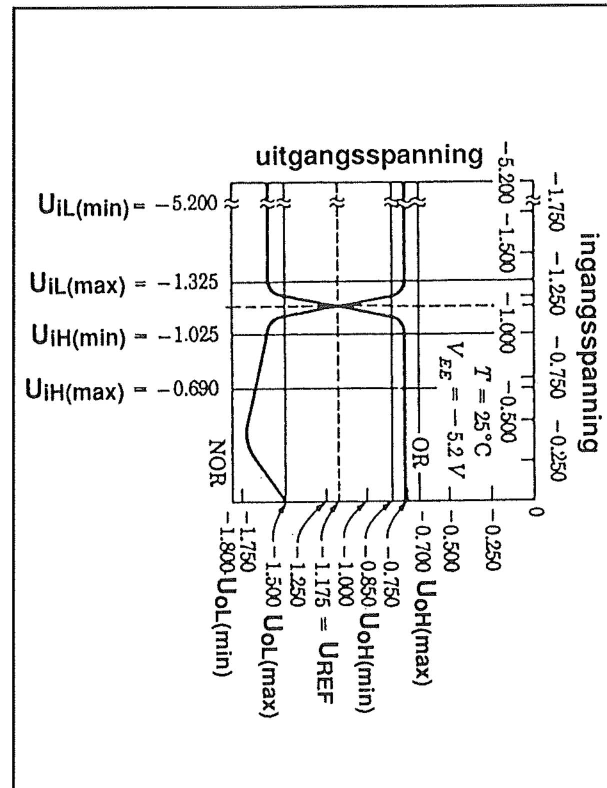
$$U_{ref} + 75\ mV = -1{,}29 + 0{,}075 = -1{,}22\ V$$

De ingangsspanning is bij logisch 1 maximaal -0,8 V.
Stoorsignalen van:

$$-0{,}8\ V - (-1{,}22\ V) = 0{,}42\ V$$

op een ingangssignaal van -0,8 V hebben dan nog net geen invloed op de uitgangstoestand.
Van deze waarde (0,42 V) is echter alleen sprake in het ideale geval, in"worst case" situaties is de storingsongevoeligheid veel minder dan 0,42 V.

In figuur 6/3.12.4-8 is een overdrachtskarakteristiek getekend van een Motorola MECL II OR poort, zoals de fabrikant die specificeert.



Figuur 6/3.12.4-8: De overdrachtskarakteristiek volgens de fabrikant.

Over het algemeen komt deze grafiek goed overeen met de berekende waarden (zie figuur 3/6.12.4-3).
Volgens de fabrikant kan de uitgangsspanning bij logisch 1 liggen tussen -0,7 V en -0,85 V en bij logisch 0 tussen -1,5 V en -1,8 V.
Een ingangsspanning positiever dan -1,025 V wordt gezien als logisch 1 en een ingangsspanning negatiever dan -1,325 V als logisch 0.
De storingsongevoeligheid is nu geen 0,42 V meer bij logisch 1, maar:

$$-0{,}85\ V - (-1{,}025\ V) = 0{,}175\ V$$

Door de erg lage storingsongevoeligheid moeten alle stoor- en ruissignalen tot een minimum beperkt worden. Om deze re-

den wordt ECL logika dan ook van een negatieve voedingsspanning voorzien.

Voedingsspanning
De voedingsspanning van bijvoorbeeld de ECL 10K serie van Motorola en Philips, moet liggen tussen -4,68 V en -5,72 V. Buiten dit bereik worden de door de fabrikant opgegeven specifikaties niet meer gehaald. De schakeling blijft echter wel correct functioneren tot een voedingsspanning van minimaal -8 V.

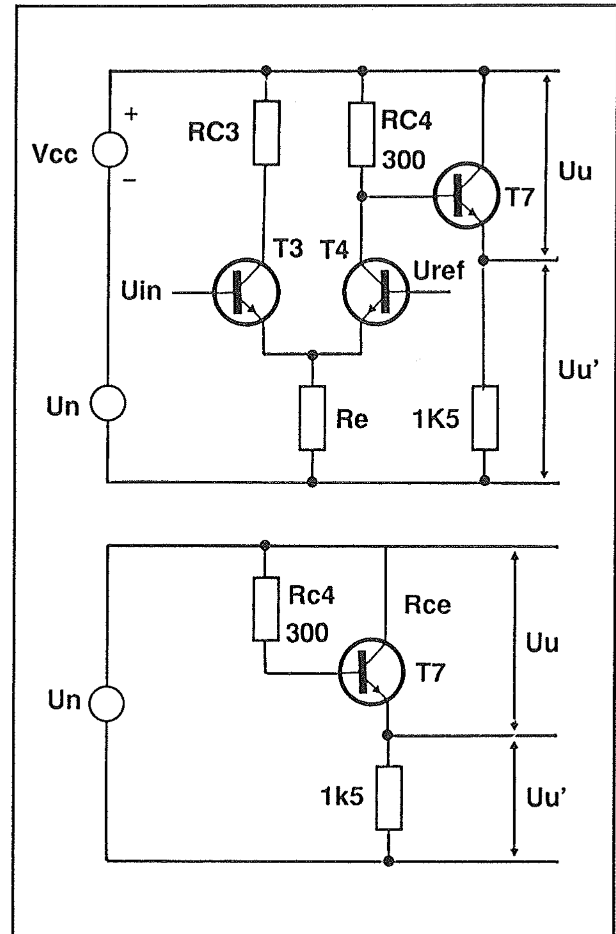
Als voedingsspanning is -5,2 V gekozen, omdat bij deze spanning de schakeling het minst gevoelig is voor storing (ruis). Om dezelfde reden, storingsongevoeligheid, is de voedingsspanning ook negatief.

De meeste stoorsignalen worden gegenereerd door stroomveranderingen. Tijdens het schakelen, van logisch 0 naar 1 of omgekeerd, verandert de stroom door de collectors van bijvoorbeeld de twee transistoren in de verschil versterker. Deze stroomverandering genereert een elektromagnetisch veld. Dit veld veroorzaakt in een ander deel van de schakeling, of in een andere poort, allerlei stoorspanningen.

Een goede benadering (simulatie) van het effect van de stoorsignalen, is om een wisselspanningsbron U_n in de voedingsspanningstoevoer te plaatsen, zie figuur 6/3.12.4-9a.

Het blijkt dat een groot deel van de spanning U_n op de uitgang $U_{u'}$ en niet op U_u staat.

Door middel van een voorbeeld zal getracht worden dit aannemelijk te maken.



Figuur 6/3.12.4-9: Een negatieve voedingsspanning verbetert de storingsongevoeligheid.

Stel dat transistor T4 gesperd is. De schakeling om U_u en $U_{u'}$ te berekenen is dan zoals in figuur 6/3.12.4-9b is getekend. Transistor T7 geleidt waardoor de collector-emitter weerstand van deze transistor als volgt is te berekenen:

$$R_{ceT7} = R_{c4} / (hFE + 1)$$

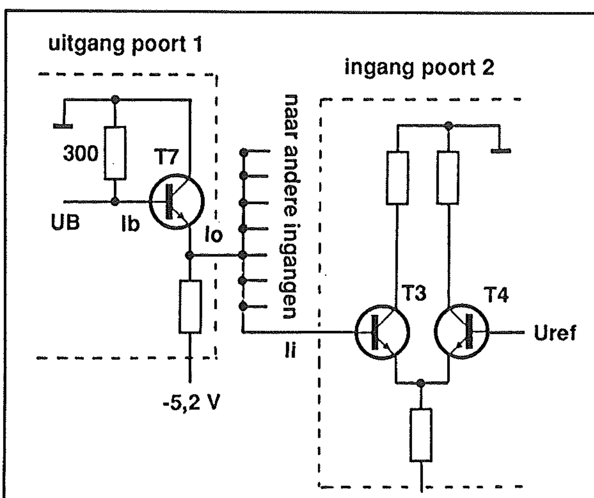
als hFE = 99 dan wordt Rce gelijk aan 3 Ω.

Over de weerstand van 1,5 kΩ ($U_{u'}$), staat nu bijna de gehele spanning U_n terwijl op de uitgang U_u vrijwel geen U_n staat.

Doordat U_u als uitgang is gekozen moet er een negatieve voedingsspanning worden aangeboden, omdat de collector van T7 verbonden moet worden met het chassis (massa), dit vanwege de noodzakelijke afscherming. Bij andere logika zit om dezelfde reden ook een uitgangspin aan de aarde.

De uitgangsbelasting
De FAN-OUT, het aantal ingangen dat op een uitgang mag worden aangesloten, is vrijwel onbeperkt als er geen rekening gehouden hoeft te worden met de storingsongevoeligheid en de schakelsnelheid.

Uit figuur 3/6.12.4-10 volgt dat er alleen uitgangsstroom geleverd hoeft te worden als de uitgang logisch 1 is. Logisch 0 betekent namelijk dat alle ingangstransistoren (T3) uit geleiding zijn waardoor er geen (ingangs)stroom geleverd hoeft te worden.



Figuur 3/6.12.4-10: Berekenen van de fan-out.

De uitgangsspanning wordt bepaald door de basis-emitter spanning en de spanning over de weerstand Rc4. Een grotere uitgangsstroom (grotere fan-out) betekent ook een grotere basisstroom I_{b7}. De spanning over Rc4 zal daardoor toenemen met als gevolg een lagere uitgangsspanning (aangenomen dat U_{be} niet verandert).

Stel dat de uitgangsspanning maximaal 0,1 V mag zakken. Wat is dan de fan-out? De uitgangsspanning zakt doordat over Rc4 spanning komt te staan. Een spanning van 0,1 V over Rc4 betekent dat er een basisstroom loopt van:

$$U_{rc4} / R_{c4} = 0{,}1\ V / 300\ \Omega = 0{,}33\ mA$$

Dit heeft een emitterstroom tot gevolg van:

$$I_e = (hFE + 1) * I_b$$
$$I_e = 100 * 0{,}33\ mA = 33\ mA$$

De uitgangsstroom I_o is de emitterstroom - de stroom door Re:

$$I_o = I_e - I_{Re7} = 33\ mA - 2{,}9\ mA = 30{,}1\ mA$$

De fan-out is te berekenen door de uitgangsstroom te delen door de ingangsstroom. De ingangsstroom van een poort is ongeveer 0,35 mA wat een fan-out geeft van:

$$I_o / I_{in} = 30{,}1\ mA / 0{,}35\ mA = 86$$

De storingsongevoeligheid wordt door het dalen van de uitgangsspanning aanzienlijk slechter, geen 0,175 V maar nog slechts 0,075 V (maximum uitgangsspanning is 0,1 V gedaald).

Invloed van de fan-out op de schakeltijden
ECL logika wordt meestal gebruikt vanwege de lage propagation delay (hoge scha-

6.12 Samenstelling en werking logika-families

kelsnelheid). De schakelsnelheid wordt echter negatief beïnvloed door een hoge fan-out, omdat elke ingang een capacitieve belasting betekent.

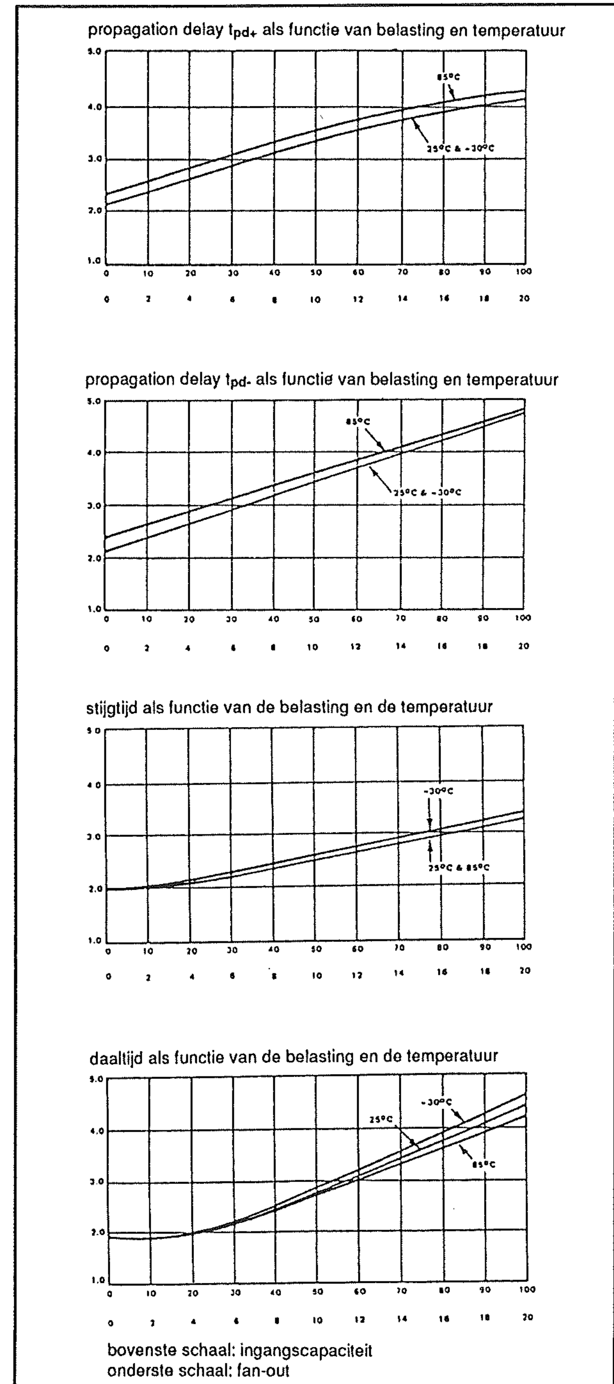
Figuur 3/6.12.4-11 geeft de stijg- en daaltijden alsmede de propagation delay bij zowel een opgaande als neergaande flank, van een MECL 10.000 poort met een uitgangsweerstand (Re7) van 50 Ω.

Deze lage uitgangsweerstand van 50 Ω is nodig omdat anders de propagation delay te snel groter wordt bij een toenemende fan-out.
Een uitgangsweerstand van bijvoorbeeld 1,5 kΩ zou betekenen dat de propagation delay met 1,5 ns per toegevoegde ingangspoort groter zou worden.

Wat bedoeld wordt met de propagation delay en de stijg- en daaltijden wordt verduidelijkt in figuur 3/6.12.4-12.

In figuur 3/6.12.4-11 is te zien dat bij een toenemende fan-out de stijgtijd minder snel groter wordt dan de daaltijd. De reden hiervan is dat tijdens het schakelen van logisch 0 naar 1 de uitgang via een weerstand van bijna 0 Ω aan aarde komt te liggen (transistor T7 geleidt), terwijl bij het schakelen van logisch 1 naar 0 de uitgang via 50 Ω aan de -5,2 V komt te liggen.
De tijd die beschikbaar is om de ingangscapaciteit op te laden of te ontladen is groter bij een serie weerstand van 50 Ω dan bij 0 Ω.

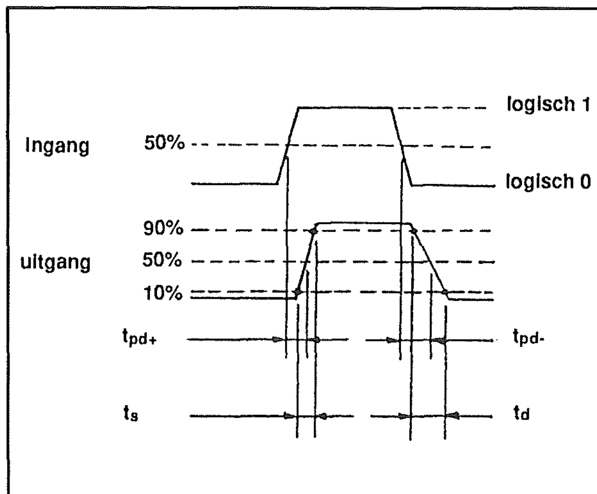
De beperkende faktor, wat betreft de fan-out, wordt bepaald door de maximaal toegestane propagation delay en niet door wat de uitgang van een poort kan leveren voordat de uitgangsspanning te ver daalt!



Figuur 3/6.12.4-11: Invloed van de belasting op de schakeltijden en de propagation delay.

Met kan besluiten dat een hoge schakelfrequentie door de capacitieve belasting, een lage fan-out betekent.

6.12 Samenstelling en werking logika-families



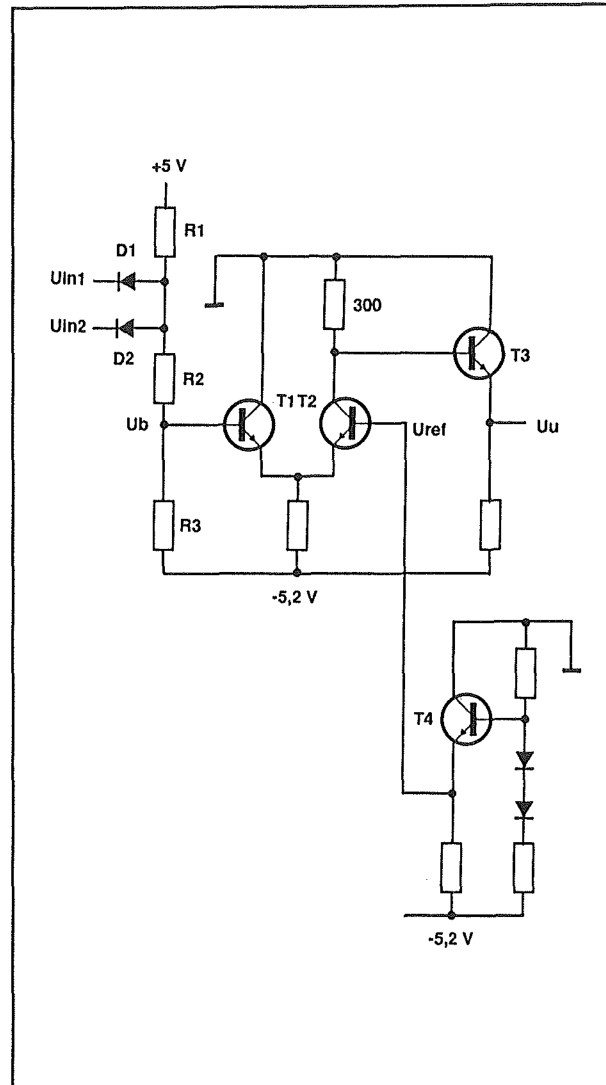
Figuur 3/6.12.4-12: De definities van de flanksteilheid en de propagation delay grafisch toegelicht.

ECL in samenwerking met andere logika

ECL logika verbruikt veel vermogen en is relatief duur, waardoor het alleen gebruikt wordt op plaatsen waar het echt noodzakelijk is. Vaak zal daarom slechts een gedeelte van de totale schakeling in ECL techniek worden uitgevoerd. Een probleem hierbij is dat de niveau's van ECL en bijvoorbeeld TTL logika nogal verschillen. Er zijn daarom speciale schakelingen die het ene niveau converteren naar het andere.

Figuur 3/6.12.4-13 geeft het schema van een TTL naar ECL converter.

De spanningsdeler R1 tot en met R3 en de dioden D1 en D2 zorgen ervoor dat de basisspanning U_b positiever is dan de referentiespanning wanneer de ingang logisch 1 (+5 V) is en negatiever dan de referentiespanning wanneer de ingang logisch 0 (0 V) is. Het resultaat is dat bij een logisch 1 (TTL-niveau) op de ingang er ook logisch 1 (ECL-niveau) op de uitgang komt te staan en omgekeerd.



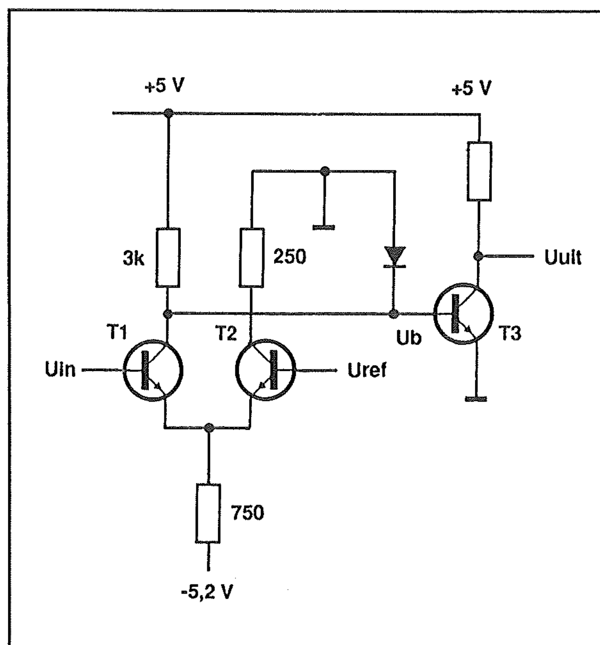
Figuur 3/6.12.4-13: Een TTL naar ECL omzetter.

Een ECL naar TTL converter is gegeven in figuur 3/6.12.4-14.

Hier zorgt logisch 1 op de ingang ervoor dat transistor T1 geleidt, waardoor op de basis van T3 een negatieve spanning komt te staan.

Transistor T3 spert en op de uitgang staat logisch 1 (TTL-niveau). Logisch 0 op de ingang betekent dat transistor T3 geleidt, waardoor er logisch 0 op de uitgang komt te staan.

6.12 Samenstelling en werking logika-families



Figuur 3/6.12.4-14: Een ECL naar TTL omzetter.

Gegevens van verschillende ECL-types

In de tabel van figuur 3/6.12.4-15 zijn een aantal parameters weergegeven van verschillende ECL-families.

	MECL 10.000			MECL III
	10.100 10.500	10.200 10.600	10.800	
JAAR VAN INTRODUCTIE	1971	1973	1976	1968
UITGANGSWEERSTAND	NEE	NEE	NEE	NEE
INGANGSWEERSTAND	JA	JA	JA	JA
MAX. INGANGSSTROOM	265 µA	410 µA	350 µA	350 µA
UITGANGSSTROOM	22 mA	22 mA	22 mA	22 mA
PIEK UITGANGSSTROOM	50 mA	50 mA	50 mA	40 mA
GELIJKSTROOOM FAN-OUT	83	54	63	68
INGANGSCAPACITEIT	2,9 pF	2,9 pF		3,3 pF
UITGANGSIMPEDANTIE	7 Ω	7 Ω	7 Ω	5 Ω
PROP-DELAY POORT	2 ns	1,5 ns	2,5 ns	1 ns
STIJGTIJD POORT	3,5 ns	2,5 ns	3,5 ns	1 ns
SCHAKELSNELHEID F-F	125 MHz	200 MHz		500 MHz
VERMOGEN PER POORT	25 mW	25 mW	2,3 mW	60 mW

Figuur 3/6.12.4-15: De belangrijkste parameters van de verschillende ECL-families.

6.12 Samenstelling en werking logika-families

3/6.12.5

Samenstelling en werking van de DTL logika

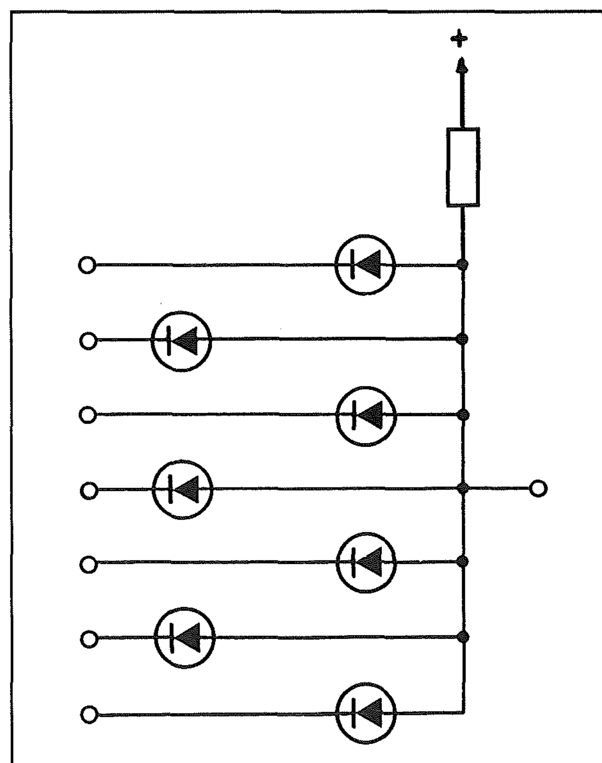
Inleiding

De DTL (Diode Transistor Logika) familie werd door Texas Instruments ontwikkeld en is de voorloper geweest van alle moderne 74xx-families. De DTL-reeks was bovendien de eerste reeks digitale schakelingen die gestandaardiseerd was opgenomen in de bekende plastic 14- en 16-pens dual-in-line (DIL) behuizingen.
Men kan zonder meer stellen dat de digitale revolutie begonnen is op het moment dat deze schakelingen op de markt verschenen! Er zijn tot zelfs in de jaren '70 gegevensverwerkende systemen, regelkringen en zelfs hele computersystemen mee gebouwd!
Thans worden DTL-schakelingen uiteraard niet meer toegepast, maar in sommige apparaten kan men deze IC's toch nog terug vinden.

Het werkingsprincipe

Het werkingsprincipe van DTL-logika zit in de naam opgesloten. De logische functies van de poorten worden tot stand gebracht door gebruik te maken van diodenetwerken. De inverterende eigenschappen ontstaan door een transistor als inverter toe te passen.

Verbindt men, zoals getekend in figuur 3/6.12.5-1, een aantal dioden met een weerstand en schakelt men deze weerstand naar een positieve voedingsspanning, dan ontstaat het meest fundamentele logische netwerk dat mogelijk is.



Figuur 3/6.12.5-1: De principiële werking van DTL logika.

De uitgang van dit netwerk zal "H" zijn als alle ingangen ook "H" zijn. De dioden sperren dan immers. Maar als één van de ingangen naar "L" gaat, dan zal ook de uitgang "L" worden. De betreffende diode gaat dan geleiden en trekt het knooppunt tussen de dioden en de weerstand naar de lage ingang.

Op deze eenvoudige manier kan men de vier basis-functies AND, NAND, OR en NOR maken. In sommige gevallen is het noodzakelijk het diode-netwerk aan te sluiten op een als inverter geschakelde transistor.

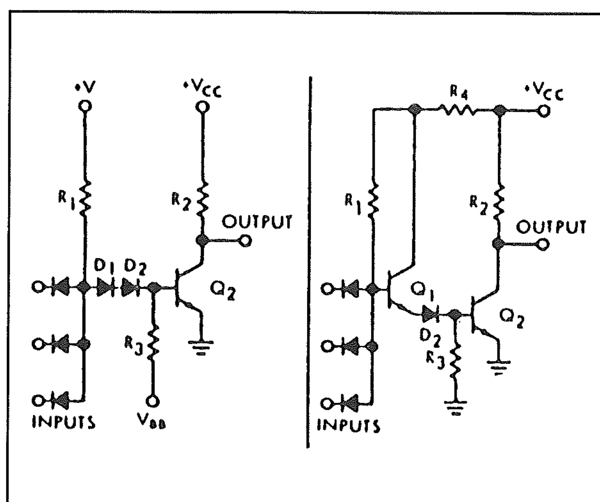
Coderingen
Texas Instruments heeft de DTL-logica in twee versies op de markt gebracht. Deze twee reeksen werden respektievelijk SN158xxN en SN159xxN genoemd, op de plaats xx komt uiteraard het betreffende typenummer.

Vanwege het grote succes van deze reeks werden de DTL-schakelingen door tal van andere fabrikanten gecloond.
De bekendste series van andere fabrikanten zijn:
- Fairchild:
  DTμL99xx (U6A99xx59)
- Motorola:
  MC8xx, MC9xx
- Philco:
  PD99xx59
- ITT:
  MIC9xx-5D
- Stewart Warner:
  SW9xx-2P
- National Semiconductor:
  DM9xxN
- Raytheon:
  RC9xxP

Uitvoeringen
Zoals reeds vermeld zijn er twee uitvoeringen:
- 158xx-serie:
  de commerciële 158xx-serie kan tussen 0 en +70 °C worden gebruikt en werd voornamelijk toegepast in massa-apparatuur waar geen al te hoge omgevingseisen aan gesteld werd;
- 159xx-serie:
  de industriële 159xx-serie heeft een uitgebreider maximaal temperatuurbereik van -55 en +125 °C.

Dit zijn de enige verschillen tussen beide reeksen, zodat men in bijna alle gevallen een IC uit de 159xx-serie kan vervangen door het equivalente type uit de 158xx-serie.

DTL-schakelingen
De door Texas op de markt gebrachte reeks DTL was niet de eerste logika die de logische functies uitvoerde door gebruik te maken van dioden.
De allereerste uitvoeringen moesten echter uit twee voedingsspanningen gevoed worden, hetgeen de praktische bruikbaarheid niet ten goede kwam.
De zogenoemde "gemodificeerde DTL-logika" die door Texas werd ontwikkeld werkt echter op één enkele positieve voedingsspanning en gebruikt een niet verzadigde transistor in plaats van één van de offset-dioden, zie figuur 3/6.12.5-2.



Figuur 3/6.12.5-2: Vergelijking tussen een conventionele DTL-schakeling (links) en de gemodificeerde schakeling van Texas (rechts).

Bij de conventionele schakeling zijn tussen het diode-netwerk aan de ingang en de basis van de inverterende transistor enige offset-dioden geschakeld.
Deze moeten er voor zorgen dat de transistor toch naar sper wordt gestuurd als alle ingangen "L" zijn.
Maar zelfs deze twee dioden waren niet voldoende om er onder alle omstandigheden van verzekerd te zijn dat aan de genoemde eis werd voldaan. Vandaar dat de basis van de transistor werd aangesloten op een negatieve voedingsspanning V_{ss} die het werkingspunt van de halfgeleider een bepaalde negatieve offset gaf. De vinding van Texas bestond er uit om een van de offset-dioden te vervangen door een transistor met een offset-diode in de emitter.

Daardoor wordt automatisch een verschuiving van het werkpunt van de halfgeleider gecreeerd zonder dat men een negatieve voedingsspanning moet ter hulp roepen.

Eigenschappen van de 15xx-families

De twee 15xx-families kenmerken zich door de volgende algemene eigenschappen:
- poortvertraging:
  25 ns
- one-shot vertraging:
  20 ns
- vermogen per poort:
  5 mW
- vermogen per flip-flop:
  20 mW
- klokfrequentiebereik:
  20 MHz
- triggerfrequentie voor flip-flop's:
  7 MHz
- voedingsspanning V_{CC}:
  5 V

Een overzicht van de functies

Toen de 158xx- en 159xx-series werden ontwikkeld kon men nog niet erg veel onderdelen op een plaatje silicium integreren. Het zal dan ook duidelijk zijn dat de DTL reeksen uit eenvoudige functies bestaan, zoals poorten en flip-flop's.
In de onderstaande lijst wordt een korte omschrijving gegeven van de schakelfunctie van de beroemdste en meest gebruikte leden uit de 158xx-familie van Texas Instruments.
- SN 15830
  2 x 4-ingang NAND/NOR
- SN 15831
  flip-flop met Set en Clear
- SN 15832
  2 x 4-ingang NAND/NOR-buffer
- SN 15833
  2 x 4-ingang expander
- SN 15834
  6 x inverter
- SN 15835
  6 x inverter
- SN 15836
  6 x inverter
- SN 15837
  6 x inverter
- SN 15844
  2 x uitbreidbare 4-in NAND/NOR power-poort
- SN 15845
  flip-flop met Set en Clear
- SN 15846
  4 x 2-ingang NAND/NOR
- SN 15848
  flip-flop met Set en Clear
- SN 15849
  4 x 2-ingang NAND/NOR
- SN 15850
  puls-getriggerde flip-flop
- SN 15851
  monostabiele multivibrator
- SN 15857

4 x 2-ingang NAND-buffer
- SN 15858
4 x 2-ingang NAND power-poort
- SN 15861
2 x uitbreidbare 4-in NAND/NOR
- SN 15862
3 x 3-ingang NAND/NOR
- SN 15863
3 x 3-ingang NAND/NOR
- SN 151800
2 x 5-ingang NAND
- SN 151801
2 x 5-ingang NAND
- SN 151802
uitbreidbare 8-ingang NAND
- SN 151803
uitbreidbare 8-ingang NAND
- SN 151804
10-ingang NAND
- SN 151805
10-ingang NAND
- SN 151810
4 x 2-ingang NOR
- SN 151811
4 x 2-ingang NOR
- SN 158093
2 x J-K Master-Slave flip-flop
- SN 158094
2 x J-K Master-Slave flip-flop
- SN 158097
2 x J-K Master-Slave flip-flop met gemeenschappelijke Clock en Clear
- SN 158099
2 x J-K Master-Slave flip-flop met gemeenschappelijke Clock en Clear

Equivalentenlijst DTL-series

In de meeste gevallen volgen de gecloonde DTL-schakelingen van andere fabrikanten dezelfde nummering als deze die door Texas Instruments wordt gehanteerd. Maar er zijn toch uitzonderingen op deze regel. Vandaar dat dit kort, bijna historisch te noemen hoofdstukje over DTL logika wordt afgesloten met een equivalentenlijst van de meest gebruikte schakelingen.

SW 705-2P	SN 158093
SW 706-2P	SN 158099
SW 708-2P	SN 158094
SW 709-2P	SN 158097
MC 830	SN 15830
MC 831	SN 15831
MC 832	SN 15832
MC 833	SN 15833
MC 834	SN 15834
MC 836	SN 15836
MC 840	SN 15835
MC 844	SN 15844
MC 845	SN 15845
MC 846	SN 15846
MC 848	SN 15848
MC 849	SN 15849
MC 850	SN 15850
MC 851	SN 15851
MC 852	SN 158099
MC 853	SN 158093
MC 855	SN 158097
MC 856	SN 158094
MC 857	SN 15857
MC 858	SN 15858
MC 861	SN 15861
MC 862	SN 15862
MC 863	SN 15863
DM 930	SN 15830
MIC 930-5D	SN 15830
RC 930	SN 15830
SW 930-2P	SN 15830
MIC 931-5D	SN 15831
DM 932	SN 15832
MIC 932-5D	SN 15832
RC 932	SN 15832
SW 932-2P	SN 15832
DM 933	SN 15833
MIC 933-5D	SN 15833
RC 933	SN 15833
SW 933-2P	SN 15833
DM 935	SN 15835

6.12 Samenstelling en werking logika-families

MC 935	SN 15838	MIC 962-5D	SN 15862
RC 935	SN 15835	RC 962	SN 15862
MIC 935-5D	SN 15835	SW 962-2P	SN 15862
SW 935-2P	SN 15835	DM 963	SN 15863
DM 936	SN 15836	MIC 963-5D	SN 15863
MIC 936-5D	SN 15836	RC 963	SN 15863
RC 936	SN 15836	SW 963-2P	SN 15863
SW 936-2P	SN 15836	MC 1800	SN 151800
DM 937	SN 15837	MC 1801	SN 151801
MIC 937-5D	SN 15837	MC 1802	SN 151802
RC 937	SN 15837	MC 1803	SN 151803
SW 937-2P	SN 15837	MC 1804	SN 151804
MC 937	SN 15837	MC 1805	SN 151805
DM 944	SN 15844	MC 1810	SN 151810
MIC 944-5D	SN 15844	MC 1811	SN 151811
RC 944	SN 15844	MC 1800	SN 151800
SW 944-2P	SN 15844	DM 9093	SN 158093
DM 945	SN 15845	MIC 9093-5D	SN 158093
MIC 945-5D	SN 15845	RC 9093	SN 158093
RC 945	SN 15845	DM 9094	SN 158094
SW 945-2P	SN 15845	MIC 9094-5D	SN 158094
DM 946	SN 15846	RC 9094	SN 158094
MIC 946-5D	SN 15846	DM 9097	SN 158097
RC 946	SN 15846	MIC 9097-5D	SN 158097
SW 946-2P	SN 15846	RC 9097	SN 158097
DM 948	SN 15848	DM 9099	SN 158099
MIC 948-5D	SN 15848	MIC 9099-5D	SN 158099
RC 948	SN 15848	RC 9099	SN 158099
SW 948-2P	SN 15848	PD 909359	SN 158093
DM 949	SN 15849	U6A 909359	SN 158093
MIC 949-5D	SN 15849	PD 909459	SN 158094
SW 949-2P	SN 15849	U6A 909459	SN 158094
MIC 950-5D	SN 15850	PD 909759	SN 158097
RC 950	SN 15850	U6A 909759	SN 158097
SW 950-2P	SN 15850	PD 909959	SN 158099
DM 951	SN 15851	U6A 909959	SN 158099
MIC 951-5D	SN 15851	PD 993059	SN 15830
RC 951	SN 15851	U6A 993059	SN 15830
SW 951-2P	SN 15851	PD 993159	SN 15831
DM 961	SN 15861	U6A 993159	SN 15831
MIC 961-5D	SN 15861	PD 993259	SN 15832
RC 961	SN 15861	U6A 993259	SN 15832
SW 961-2P	SN 15861	PD 993359	SN 15833
DM 962	SN 15862	U6A 993359	SN 15833

6.12 Samenstelling en werking logika-families

PD 993559	SN 15835	U6A 994859	SN 15848
U6A 993559	SN 15835	PD994959	SN 15849
PD 993659	SN 15836	U6A 994959	SN 15849
U6A 993659	SN 15836	PD 995059	SN 15850
PD 993759	SN 15837	U6A 995059	SN 15850
U6A 993759	SN 15837	PD995159	SN 15851
PD 994459	SN 15844	U6A 995159	SN 15851
U6A 994459	SN 15844	PD996159	SN 15861
PD 994559	SN 15845	U6A 996159	SN 15861
U6A 994559	SN 15845	PD996259	SN 15862
PD994659	SN 15846	U6A 996259	SN 15862
U6A 994659	SN 15846	PD996359	SN 15863
PD 994859	SN 15848	U6A 996359	SN 15863

3/6.12.6

Samenstelling en werking van de BI(C)MOS logika

Inleiding

Integratie!
BI(C)MOS is de verzamelnaam van een aantal technologieën, waarbij men er in slaagt de oude bipolaire transistoren en de moderne CMOS-schakelingen samen te integreren op één chip. Deze integratie heeft een aantal voordelen, die voornamelijk van belang zijn bij interface-schakelingen in zeer snelle computersystemen en in bijvoorbeeld ADC's en DAC's. De gunstige eigenschappen van bipolaire transistoren kunnen dan gecombineerd worden met de voortreffelijke eigenschappen van CMOS.

Het probleem van gecombineerde schakelingen
Op geïntegreerd niveau bestaat er een grote scheiding tussen analoge en digitale elektronica. Analoge schakelingen worden meestal vervaardigd volgens het bipolaire principe en soms in MOS. Daarbij worden "normale" transistoren geïntegreerd. Digitale schakelingen werken tegenwoordig meestal volgens het CMOS-principe. De logische schakelfuncties worden uitgevoerd door CMOS-transistoren in serie, in cascade of parallel te schakelen.
Met de steeds stijgende integratie probeert men technologieën te ontwikkelen, waarmee het zonder al te grote problemen mogelijk is analoge en digitale schakelingen met elkaar te kombineren. Tot voor kort gebeurde dat hoofdzakelijk middels CMOS-technologie. Deze technologie heeft echter een aantal nadelen, nadelen die opgelost zouden kunnen worden als men van de oude, vertrouwde bipolaire techniek gebruik zou kunnen maken.

Het probleem van de interfaces
In computer systemen wordt veel gebruik gemaakt van zogenoemde interfaces. Deze staan bijvoorbeeld tussen de data- en adres-bussen en de perifere I/O-schakelingen. Door het stijgen van de verwerkingssnelheid van de systemen worden aan deze interfaces steeds hogere eisen gesteld, zeker wat betreft schakelsnelheid.
De traditionele busschakelingen, uitgevoerd in CMOS-techniek, laten het dan afweten. Dit is een gevolg van het feit dat CMOS-schakelingen een beperkte stroomcapaciteit hebben, waardoor de paracitaire capaciteiten van de belasting een te grote rol op de stijg- en daaltijden van de signalen gaan hebben.

Maar bovendien blijkt uit analyses dat in computersystemen 30 % van het totale vermogen in busdrivers wordt gedissipeerd.

Bij bussen zijn echter altijd slechts twee modulen tegelijk aktief: de zender en de ontvanger. Alle andere aangesloten apparatuur moet dan in een niet-aktieve toestand worden gehouden.
Met gecombineerde bipolaire en CMOS schakelingen kunnen bus-drivers worden gemaakt die in niet-aktieve toestand (uitgangen in de hoog-impedante 3-state toestand) uiterst weinig vermogen opnemen.

Bipolair contra CMOS
Hoewel met de 1 µm advanced CMOS technologieën (zoals ACL) poortvertragingstijden van 3 ns mogelijk zijn, wordt met het bipolaire 2 µm-proces dezelfde snelheid gehaald.
Bipolaire schakelingen zijn op hetzelfde technologieniveau (afmetingen) dus tweemaal zo snel als CMOS-componenten. Verder hebben bipolaire schakelingen nog het voordeel dat ze grote stromen kunnen verwerken (een bipolaire transistor kan vijf maal zoveel stroom schakelen als een even grote MOS-transistor). Bovendien kunnen sommige schakelingen moeilijk in CMOS worden gerealiseerd of is het zelfs geheel onmogelijk om dat te doen. Een zeer precieze differentiële versterker, bijvoorbeeld, die in A/D-omzetters of leesversterkers wordt gebruikt, heeft nauwkeurige en stabiele referentiespanningen nodig en kan alleen betrouwbaar met bipolaire processen (band-gap referentie) worden gemaakt.

Conclusie
Door de beste eigenschappen van beide technologieën te combineren kan dus een bijna ideale schakeling ontstaan. De geringe dissipatie en hoge dichtheden van CMOS kunnen gecombineerd worden met de hoge snelheid en grote stromen van het bipolaire proces!

BI(C)MOS
Door diverse fabrikanten zijn geheel nieuwe families van geïntegreerde schakelingen ontwikkeld, waarin zowel van bipolaire als van CMOS-technologieën wordt gebruik gemaakt.
Deze families worden samengevat door de soortnamen BIMOS of BICMOS.
Op dit moment zijn drie families leverbaar:
- BiCMOS Bus Interface Logic 74BCTxxx van Texas Instruments;
- BiMOS Interface Logic 74FCTxxx van Harris;
- BiCMOS Advanced Interface Logic 74ABTxxx van Philips.

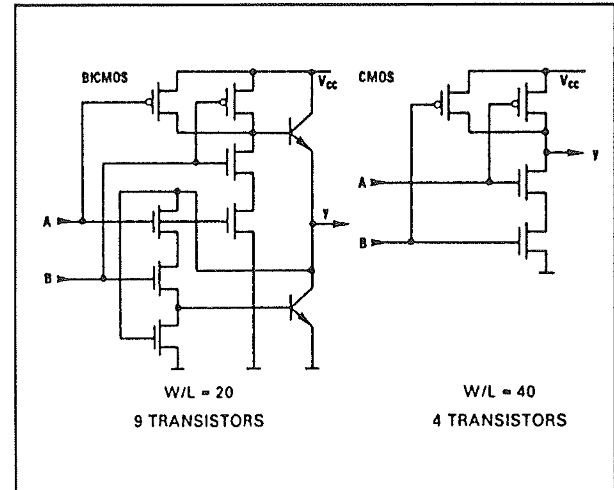
BiCMOS Bus Interface Logic 74BCTxxx

Inleiding
Door Texas Instruments is een nieuwe serie bus-interface schakelingen ontwikkeld voor "backplanes" en zwaarcapacitieve belastingen.
Met zeer hoge snelheid kunnen in het commerciële temperatuurgebied uitgangsstromen tot 24 of 64 mA worden geschakeld. In deze 74BCTxxx-serie is een groot aantal, op bussen georiënteerde functies, zoals latches, buffers, drivers en transceivers verkrijgbaar.

Werking
In figuur 3/6.12.6-1 wordt een CMOS-poort vergeleken met dezelfde poort in BiCMOS-technologie.
Hoewel in de gemengde schakeling aanzienlijk meer componenten worden gebruikt (9 transistoren in plaats van 4) heeft dit slechts weinig effect op de beno-

digde ruimte. Door de grote aandrijfcapaciteit van de bipolaire transistoren kunnen alle elementen aanzienlijk worden verkleind.
Aangezien bij zeer snelle schakelingen de lijnlengten (in het mm-gebied) belangrijker worden dan de poortvertragingstijden zal de BiCMOS-technologie vooral bij ingewikkelde IC's (met veel onderlinge verbindingen op de chip zoals bijvoorbeeld ASIC's) een steeds grotere rol spelen. Bij logische schakelingen wordt in de BiCMOS-technologie zoveel mogelijk gebruik gemaakt van CMOS om het gedissipeerde vermogen te beperken, terwijl bipolaire transistoren in de uitgangstrappen voor de benodigde uitgangsstroom zorgen. Het zal duidelijk zijn dat dit alleen nut heeft voor complexe logische schakelingen. BiCMOS wordt dan ook niet toegepast bij kleine, eenvoudige poorten, buffers en inverters.
In figuur 3/6.12.6-2 worden de vereenvoudigde schema's van een conventionele bipolaire uitgangstrap en een BiCMOS-versie vergeleken.

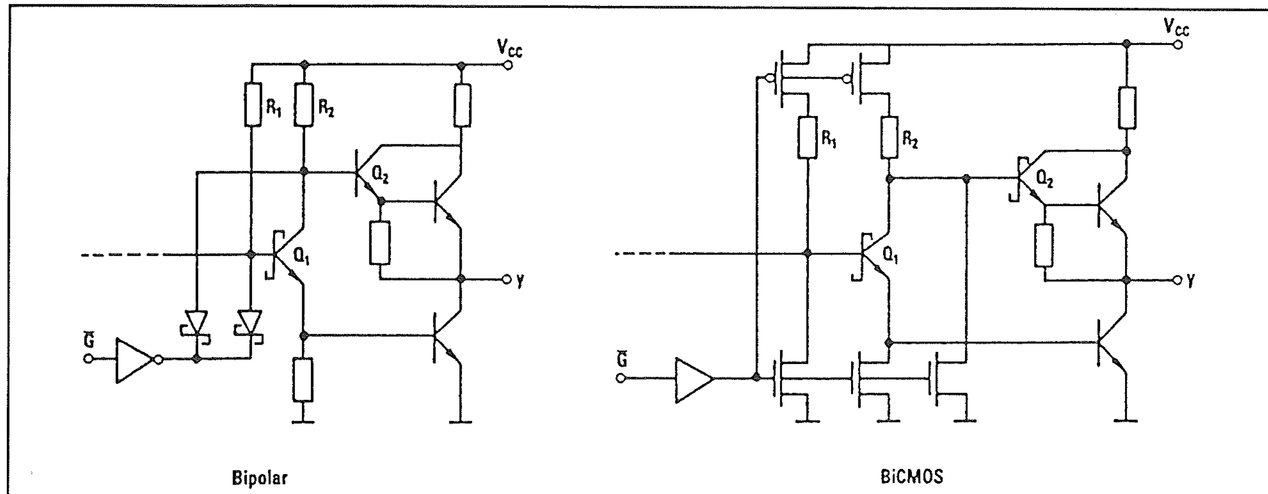
Om de bipolaire uitgangstrap hoogimpedant te maken moeten de bases van Q1 en Q2 met de enable-lijn op een laag potentiaal worden gebracht. Doordat dan bijna de gehele voedingsspanning over de weerstanden R1 en R2 staat, kan het zelfs gebeuren dat de busdriver in niet-aktieve toestand meer vermogen dissipeert dan in aktieve toestand.
In de BiCMOS-schakeling zijn p-kanaal MOS transistoren in serie met R1 en R2 geschakeld. Nu wordt de uitgang hoogimpedant als een hoge spanning op de enable-lijn komt, zodat de voedingsspanning voor de uitgangstrap inwendig wordt afgeschakeld. Alleen de ingangstrap voor de enable-logika blijft hier aktief.



Figuur 3/6.12.6-1: Vergelijking van een poortschakeling in BiCMOS (links) en CMOS (rechts).

BiCMOS in de praktijk
Aan de hand van figuur 3/6.12.6-3 wordt geïllustreerd hoe het gebruik van BiCMOS-schakelingen de vermogensbalans van een systeem gunstig kan beïnvloeden. De 6 modulen worden via 74ALS245, 74F245 of 74BCT245 transceivers op de bus aangesloten.

In de tabel van figuur 3/6.12.6-4 wordt een overzicht gegeven van de belangrijkste karakteristieken van de drie families. Die van de 74BCT245 en de 74F245 zijn vrijwel identiek. De ingangsstromen van de ALS-schakeling zijn iets kleiner. Alle drie zijn dus zeer geschikt voor de aandrijving van laag-impedante buslijnen.
Het grote voordeel van BiCMOS komt tot uiting in de drastische vermindering van het opgenomen vermogen (zie ook de tabel in figuur 3/6.12.6-5).
Zoals al werd opgemerkt zijn slechts twee schakelingen tegelijk aktief, terwijl de overige vier zijn afgeschakeld. De ALS-schakelingen dissiperen in niet-aktieve toestand (I_{CCZ}) iets meer dan in aktieve toestand.



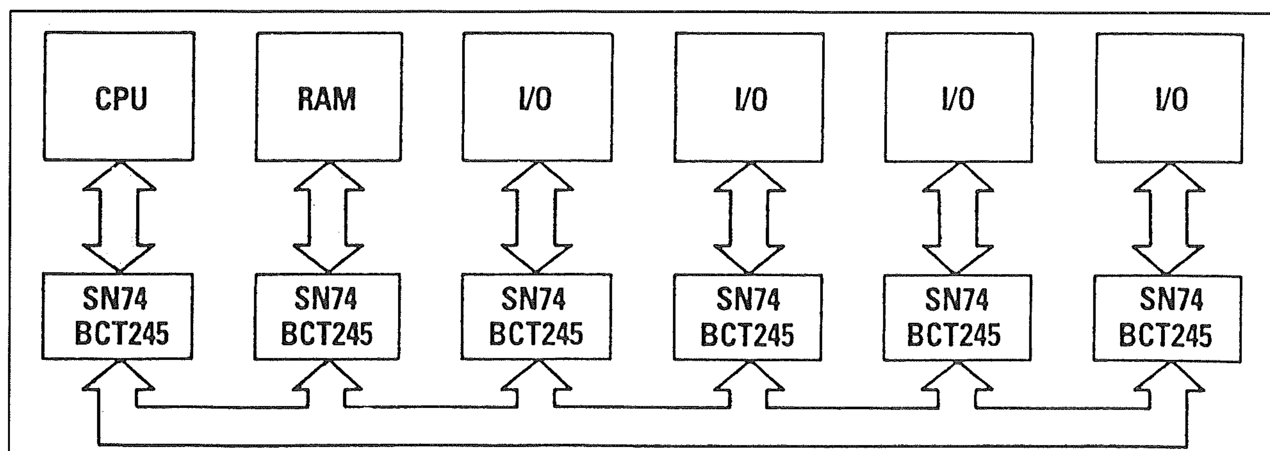
Figuur 3/6.12.6-2: 3-state uitgangstrap in bipolaire (links) en BiCMOS-technologie (rechts).

De F-schakelingen dissiperen in beide gevallen ongeveer even veel, maar dan wel tweemaal zoveel als de ALS-drivers. De BiCMOS-drivers dissiperen in niet-aktieve toestand aanzienlijk minder dan in aktieve toestand, hetgeen resulteert in een veel kleinere totaalstroom.

Algemene eigenschappen

De algemene eigenschappen van de BiCMOS-schakelingen van Texas Instruments kunnen als volgt worden samengevat:

- alleen bus-interface functies;
- dezelfde aansluitingen als 74xx-serie;
- veel kleinere vermogensdissipatie in hoog-impedante 3-state toestand;
- geïntegreerde beveiliging tegen elektrostatische spanningen;
- hoog-impedante in- en uitgangen bij power-down;
- poortvertraging gemiddeld 7,2 ns;
- klokfrequentiebereik maximaal 120 MHz;
- voedingsspanning 4,5 tot 5,5 V;
- uitgangs sink/source-stroom 24 of 64 mA.



Figuur 3/6.12.6-3: Een eenvoudig bus-systeem, waarbij BiCMOS-schakelingen worden toegepast als I/O-drivers.

6.12 Samenstelling en werking logika-families

	SN74ALS245A	SN74F245	SN74BCT245
I_{IH} at V_{IH} = 2.7 V	0.02 mA	0.07 mA	0.07 mA
I_{IL} at V_{IL} = 0.5 V	−0.1 mA	−0.65 mA	−0.65 mA
I_{OH} at V_{OH} = 2.05 V	−15 mA	−15 mA	−15 mA
I_{OL} at V_{OL} = 0.5 V	24 (48) mA	64 mA	64 mA
t_{pd} (max)	10 ns	7.0 ns	7.2 ns

Figuur 3/6.12.6-4: Vergelijking van de elektrische in- en uitgangskenmerken van drie logische families.

	SN74ALS245A	SN74F245	SN74BCT245
$I_{CC^*)}$	2 x 50 mA	2 x 105 mA	2 x 73.5 mA
I_{CCZ}	4 x 58 mA	4 x 110 mA	4 x 15 mA
Total I_{CC}	332 mA	650 mA	207 mA

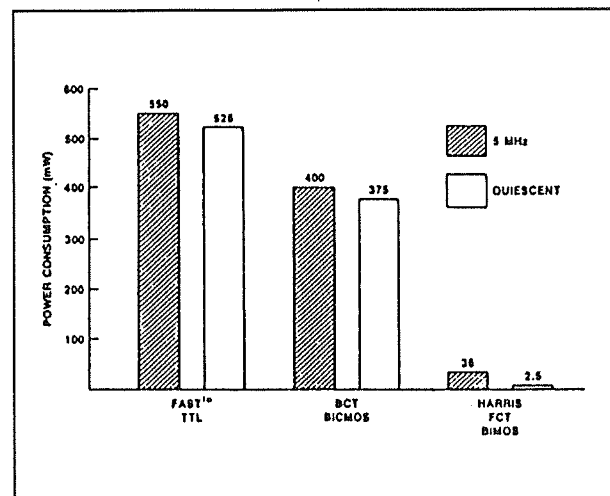
Figuur 3/6.12.6-5: Vergelijking van het opgenomen vermogen bij toepassing van de families in figuur 3/6.12.6-3.

BiMOS Interface Logic 74FCTxxx

Inleiding

Ook door Harris is een nieuwe serie BiMOS logika voor TTL backplane-interface doeleinden ontwikkeld, namelijk de 74FCTxxx-familie. Met IC's uit deze gemengd bipolaire/CMOS0-familie zijn zelfs nog grotere besparingen van het opgenomen vermogen mogelijk dan met de BCT-familie. Een en ander wordt grafisch verduidelijkt in figuur 3/6.12.6-6.
Met deze familie worden bus-rivaliteit en schakelruis (vooral in VME-bussen of soortgelijke) vermeden. De snelheid van de FCT-familie komt overeen met die van bipolaire FAST-typen, terwijl sink-uitgangsstromen van 48 tot 64 mA toelaatbaar zijn. Dit is voldoende voor een betrouwbare aansturing van volledig bezette bussen, zoals de 21-slots VME-bus.

De 74FCTxxx-componenten voldoen aan de eisen die in de JEDEC industriestandaard nr. 18 zijn vastgesteld.



Figuur 3/6.12.6-6: Vergelijking van de opgenomen vermogens van een 8-voudige transceiver in FAST, BCT en FCT.

Het belangrijkste verschil tussen de FCT-serie en de beide concurrerende families

(FAST en BCT) wordt gevormd door het opgenomen vermogen. FAST en BCT nemen bij 5 MHz in niet-aktieve toestand ongeveer 150 maal zoveel vermogen op en in aktieve toestand circa 10 maal zoveel. Bovendien heeft FCT als kenmerk dat bus-rivaliteit door ongelijke sink- en sourcestromen wordt voorkomen. Door het ontbreken van clamp-dioden naar de positieve voeding worden grote stromen bij power-down vermeden. Tevens is het hierdoor mogelijk om prints te verwisselen terwijl de voedingsspanning aan staat.

Algemene eigenschappen

De algemene eigenschappen van deze Harris schakelingen:

- alleen bus-interface functies;
- dezelfde aansluitingen als 74F/ASxxx-serie;
- ultra-lage, zuivere CMOS vermogensdissipatie tijdens bedrijf en bijna nul standby-vermogen;
- minimale EMI en RFI uitstraling;
- sink en source uitgangsstromen voldoen aan VME, multibus en andere standaards;
- poortvertraging typisch 3,5 ns;
- klokfrequentiebereik maximaal 120 MHz;
- voedingsspanning 5 V +/-5 %;
- uitgangsstroom sink: 70 mA maximaal;
- uitgangsstroom source: -30 mA maximaal;

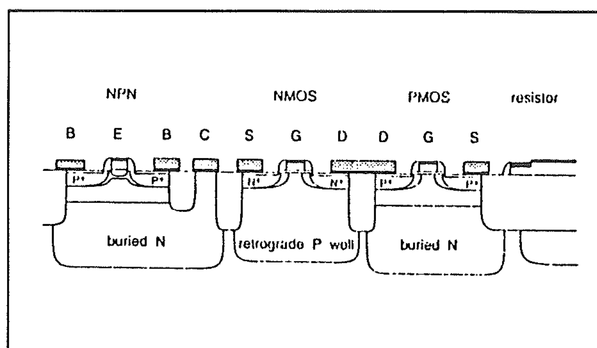
Advanced BiCMOS Interface Logic 74ABTxx

Inleiding

Met behulp van het QUBiC-proces is door Philips de hoeksteen gelegd voor een nieuwe familie BiCMOS bus-interface schakelingen op TTL-niveau. De 74ABTxx-familie (Advanced BiCMOS Technologie) combineert de beste eigenschappen van CMOS en bipolaire processen en bestaat in principe uit 8-voudige latches, buffers, drivers en transceivers. Door Texas Instruments worden als "second-source" ABT-schakelingen gemaakt met behulp van het EPIC-2B proces.

Terwijl andere BiCMOS-processen in feite modificaties van bestaande bipolaire of CMOS processen zijn, is QUBiC een werkelijke vermenging van beide technologieën (figuur 3/6.12.6-7).

Met QUBiC kunnen N- en P-kanaal MOS transistoren, NPN- en laterale PNP-transistoren, Schottky-, standaard- en zenerdioden, zekeringen en weerstanden worden vervaardigd. Hierdoor zijn CMOS-, TTL- en ECL-signaalniveaus op dezelfde chip toegestaan. QUBiC is daarom niet alleen geschikt voor ABT-logika, maar ook voor PLD's, gate array's, ASIC's, geheugens en schakelingen voor Future-Bus+ toepassingen.



Figuur 3/6.12.6-7: Dwarsdoorsnede van een ABT-schakeling waarin de belangrijkste elementen worden getoond die met het QUBiC-proces gemaakt kunnen worden.

ABT is, wat betreft uitgangsstroom, snelheid en opgenomen vermogen, de beste

keuze voor logische toepassingen. Voor eenvoudiger functies, zoals poorten, Schmitt-trigger's, flip-flop's, multiplexers en tellers kan ABT uitstekend worden aangevuld met ACL.

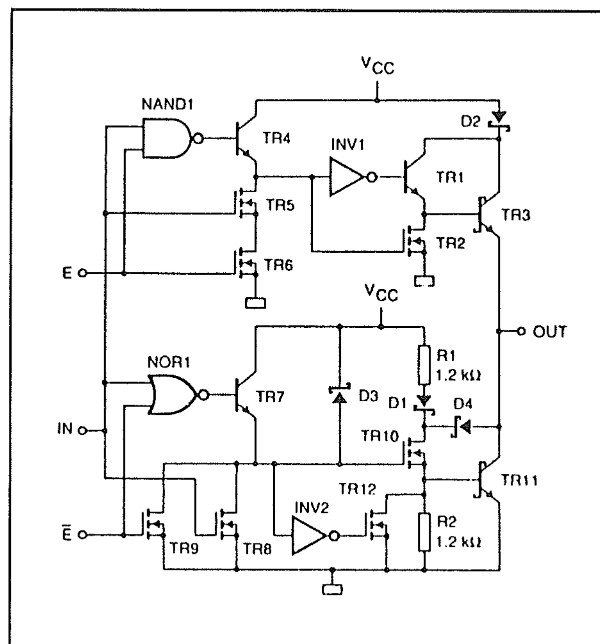
Basis 3-state cel

In figuur 3/6.12.6-8 is de basiscel getekend die in de meeste ABT 3-state busdriver-schakelingen wordt toegepast. Deze oplossing maakt uiterst kleine interne afmetingen mogelijk waarbij zeer korte vertragingstijden en geringe dissipaties optreden. De andere delen van een ABT-schakeling zijn gewoonlijk eenvoudiger.

Algemene eigenschappen

De algemene eigenschappen van deze nieuwe Philips logica-reeks zijn als volgt samen te vatten:

- alleen bus-interface functies;
- dezelfde aansluitingen als 74xx-serie;
- zeer geringe vermogensdissipatie;
- uitgangsstroom sink +64 mA;
- uitgangsstroom source -32 mA;
- ESD-beveiliging maximaal 2000 V;
- poortvertraging 2,9 ns typisch;
- klokfrequentiebereik 120 MHz maximaal;
- voedingsspanning 5 V +/-10 %.



Figuur 3/6.12.6-8: De fundamentele samenstelling van een 3-state ABT-cel.

6.12 Samenstelling en werking logika-families

3/6.13.6 FIFO geheugens

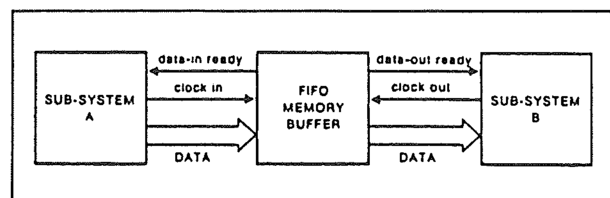
Inleiding

Het komt vaak voor dat verschillende digitale systemen met verschillende snelheden werken. In feite is het veiliger ervan uit te gaan dat dit altijd het geval is. Microprocessoren en bijbehorende geheugens werken bijvoorbeeld veel sneller dan andere componenten, zoals het toetsenbord en de floppy disk. Wanneer nu informatie van het ene (sub)systeem naar het andere moet worden verplaatst, wordt de snelheid daarvan beperkt door het langzaamste systeem. Het snelle systeem moet telkens even wachten totdat het langzame de data heeft overgenomen. Wanneer dat niet mogelijk of niet gewenst is, moet het snelheidsverschil worden overwonnen door een tussenliggende databuffer. Bij parallel transport moet de buffer het aantal bits per woord kunnen opnemen. Bovendien moet hij lang genoeg zijn om alle woorden die in een bepaalde periode moeten worden getransporteerd (een data-blok) te kunnen bevatten. Wanneer de gegevens die het eerst in de buffer gaan er aan de andere kant ook weer als eerste uit komen, spreekt men van een FIFO (First-In First-Out) geheugen of register. De data schuift daarbij steeds zo ver mogelijk door naar achteren. Het eerste woord komt dus helemaal achteraan, het tweede woord komt één plaats minder ver terecht, enzovoorts. De FIFO werkt asynchroon als het laden aan de ene kant geheel onafhankelijk van het lossen aan de andere kant kan gebeuren. Men kan zich een FIFO ook voorstellen als een regenpijp waar aan de bovenkant tennisballen ingegooid worden, terwijl iemand anders ze er aan de onderkant weer uithaalt.

Basissignalen

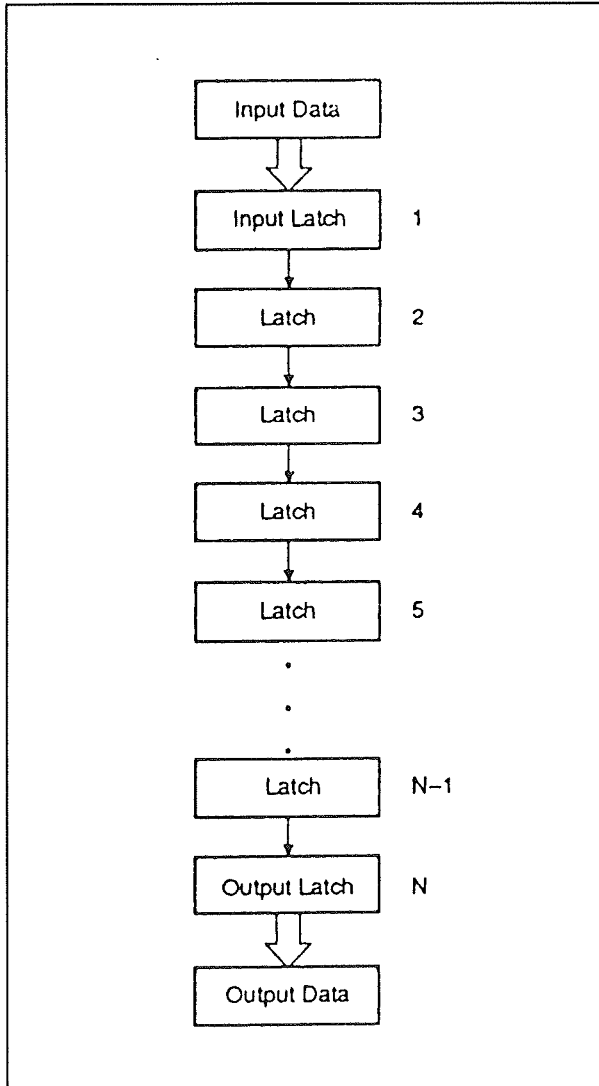
Zoals ieder geheugen wordt ook een FIFO bestuurd door een aantal signalen.

- DIR
  Zo moet de verzendende kant weten of er nog plaats is in de FIFO. Aan de ingang is dan ook een besturingssignaal aanwezig dat dit aangeeft. Dit signaal "Data-In Ready" (DIR) is meestal HOOG als er nog plaats is. Als de FIFO vol is, gaat dit signaal LAAG en blijft LAAG totdat er data van de uitgang wordt afgenomen.
  De laatste plaats komt dan vrij en alle overblijvende data schuift dus tegelijk één plaats naar achteren op en DIR wordt weer HOOG.
- DOR
  Aan de uitgang van de FIFO is een besturingssignaal beschikbaar dat aangeeft of de buffer nog data bevat. Dit signaal "Data-Out Ready" (DOR) is meestal HOOG als er nog data aanwezig is. Is de FIFO leeg dan gaat DOR LAAG en blijft LAAG totdat er weer data (aan de uitgang) aanwezig is.
- Clock
  Het clocksignaal waarmee de data wordt ingeschreven wordt Shift-In, Clock-In of Load Clock genoemd. De data wordt uitgelezen met Shift-Out, Clock-Out of Unload Clock (zie figuur 3/6.13.6-1).



Figuur 3/6.13.6-1: Bij datatransport tussen twee systemen moet het verschil in snelheid worden opgevangen door een asynchrone buffer.

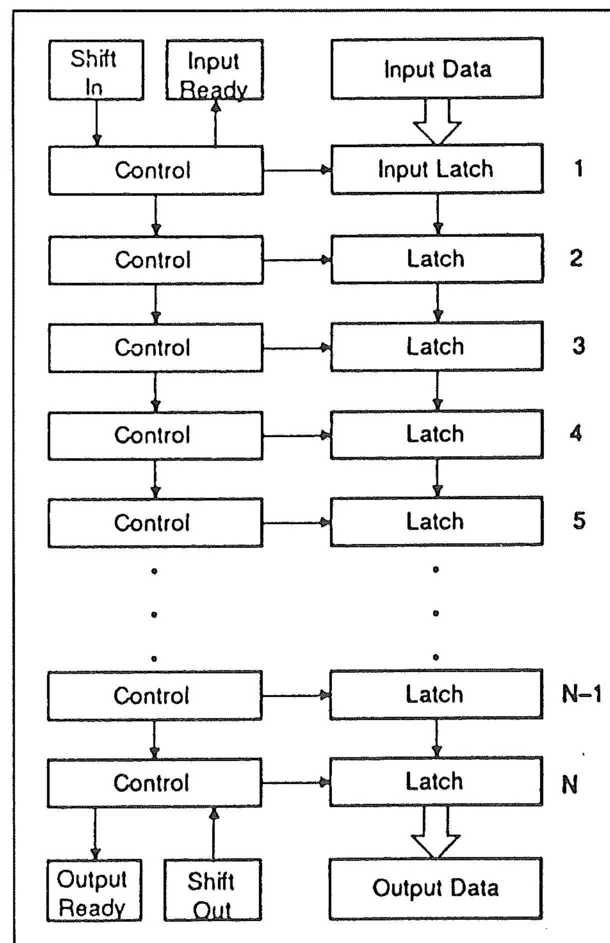
6.13 Digitale geheugens



Figuur 3/6.13.6-2: Opbouw van een FIFO (met vaste lengte) door middel van achter elkaar geplaatste latches.

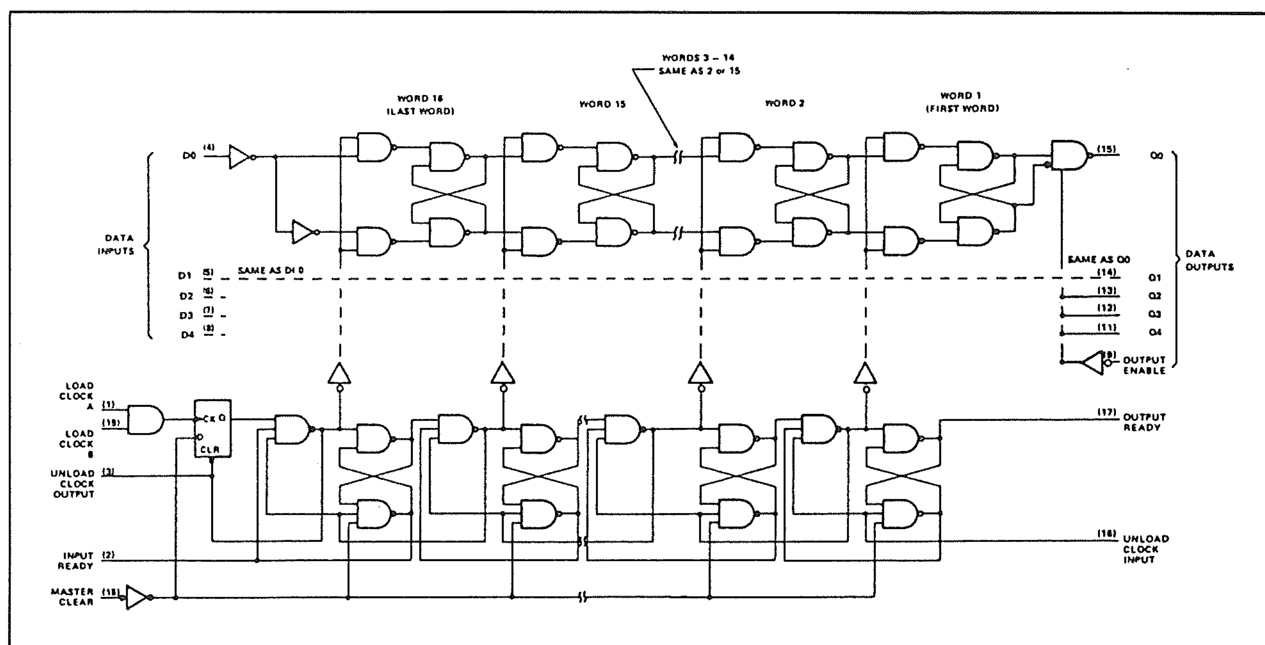
De principes van een FIFO

Voor het transporteren van de data binnen de FIFO is geen clock nodig. Van de buitenwereld bekeken schuift de data vanzelf door naar achteren. Het spreekt vanzelf dat FIFO's op verschillende manieren gebouwd kunnen worden.
De eerste gedachte is uit te gaan van een schuifregister of een aantal achter elkaar geplaatste latches. In deze vorm "valt" de data echter niet door naar achteren, maar schuiven alle woorden telkens één plaats tegelijk op. Is zo'n register bijvoorbeeld N woorden lang, dan zijn N klokpulsen nodig om de data weer aan de uitgang te laten verschijnen. De FIFO heeft een vaste lengte (figuur 3/6.13.6-2). Bovendien is na opstarten de informatie aan de uitgang pas geldig vanaf de N-de clockpuls, omdat de data daarvoor willekeurig en onbekend is. Deze FIFO heeft ook het nadeel dat niet met twee verschillende snelheden aan ingang en uitgang kan worden gewerkt.

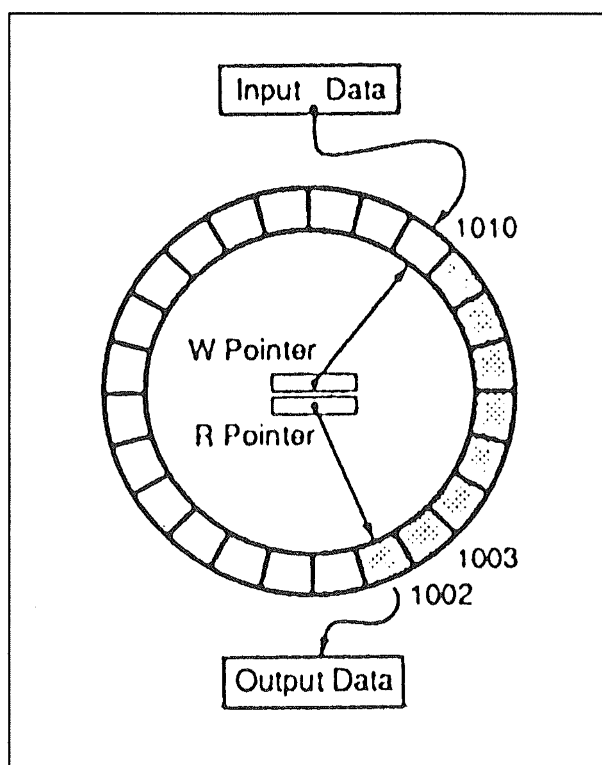


Figuur 3/6.13.6-3: Principiële opbouw van een FIFO met variabele lengte.

6.13 Digitale geheugens



Figuur 3/6.13.6-4: Functioneel blokschema van een uit latches en flip-flop's samengestelde FIFO met variabele lengte (type 74S225).

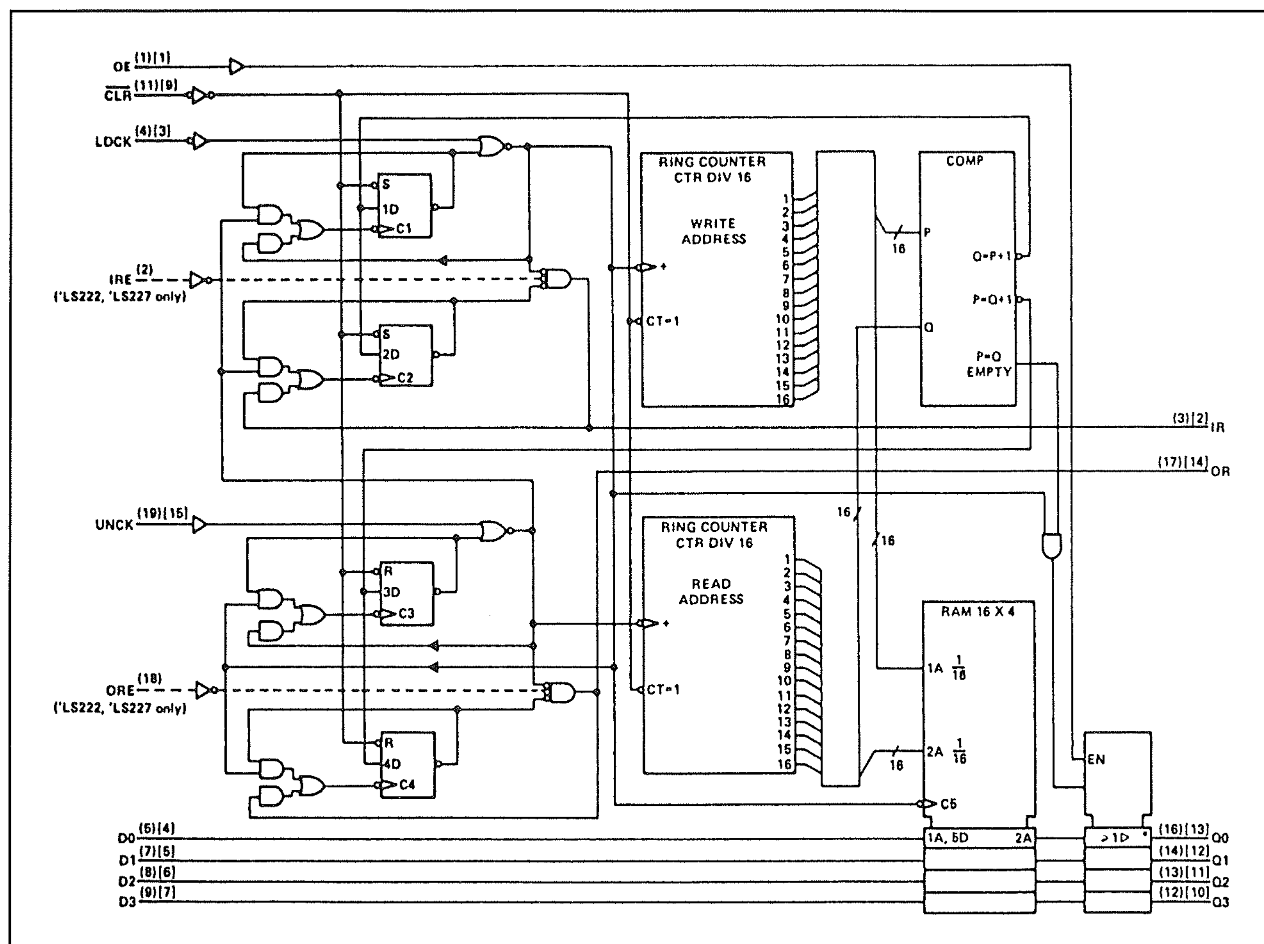


Figuur 3/6.13.6-5: Cirkelvormige geheugenstructuur van een FIFO met twee pointers.

Een betere oplossing is naast de rij latches een reeks besturingseenheden op te nemen (zie figuur 3/6.13.6-3). De latches bevatten dan net als in figuur 3/6.13.6-2 de data. De besturingseenheden leveren nu "vlaggen" die aangeven of de bijbehorende latches geldige data bevatten of niet. Van bovenaf komende data kan dan telkens "zinken" tot in de latch die zich boven de laatst gevulde latch bevindt. Op deze wijze ontstaat dus een register met variabele lengte. In figuur 3/6.13.6-4 is getekend hoe een dergelijke FIFO er in de praktijk uitziet.

Een efficiëntere manier om een FIFO op te bouwen is gebruik te maken van een opslag-array dat zo breed is als de data en daarbij twee "pointers" toe te passen. De ene pointer wijst de lokatie aan waar nieuwe data naar toe wordt geschreven, terwijl de tweede pointer bijhoudt waar de data moet worden uitgelezen.

6.13 Digitale geheugens



Figuur 3/6.13.6-6: Functioneel blokschema van een asynchrone FIFO met ingebouwde RAM, ringtellers en comparator.

Wanneer één van beide wordt gebruikt om toegang tot een lokatie te verkrijgen, wordt die automatisch met één verhoogd. Heeft een pointer de laatste positie in het array bereikt dan zal een volgende verhoging tot gevolg hebben dat de pointer op het begin van het array terecht komt. Deze structuur ziet er dus uit als een gesloten lus met twee pointers (figuur 3/6.13.6-5) en kan ook met software alleen worden uitgevoerd.

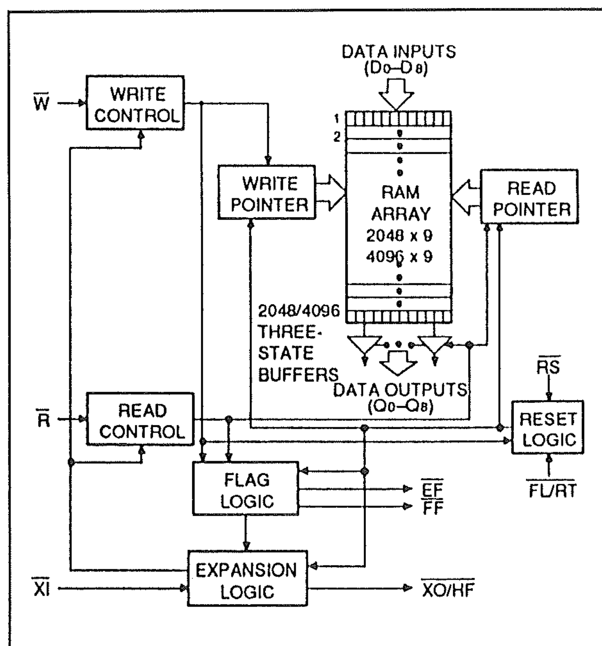
Deze benadering resulteert in een veel kortere "doorvaltijd", terwijl de lengte toch variabel blijft. In figuur 3/6.13.6-6 is getekend hoe dit met een statische RAM wordt gedaan (alle onderdelen bevinden zich op één chip). De RAM is voorzien van een dubbele adressering: één voor de ingang en één voor de uitgang. Het is dus een "dual port" geheugen, echter met de beperking dat lezen en schrijven niet verwisselbaar zijn. De bovenste ringteller houdt het laadadres bij en de onderste het uitleesadres. De comparator vergelijkt beide adressen en levert daardoor het Input-Ready en het Output-Ready-signaal. Bij deze FIFO-typen wordt door een LAGE IR of OR ook de laad- of de ontlaadclock gesperd. De bovenste ringteller en de RAM worden tegelijk door het LDCK-signaal geklokt.

6.13 Digitale geheugens

Figuur 3/6.13.6-7 toont het vereenvoudigde blokschema van een moderne, grotere FIFO die op hetzelfde principe werkt. Er zijn nu drie statussignalen:

- FIFO leeg (empty, $\overline{EF}$) die met Output-Ready overeenkomt;
- half vol ($\overline{HF}$);
- vol (full, $\overline{FF}$) die bij andere typen ook wel Input-Ready wordt genoemd.

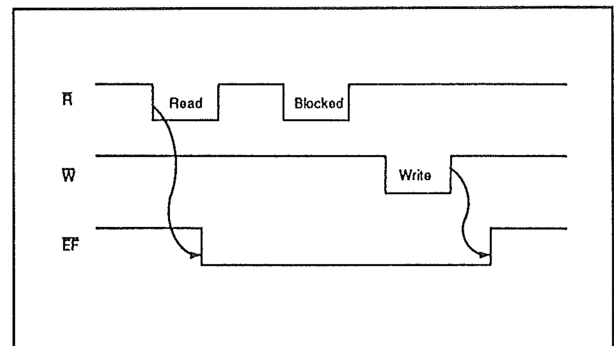
Met behulp van de $\overline{XI}$- en $\overline{XO}$-pennen kan de FIFO onbeperkt worden uitgebreid, terwijl de doorvaltijd 50 ns blijft.



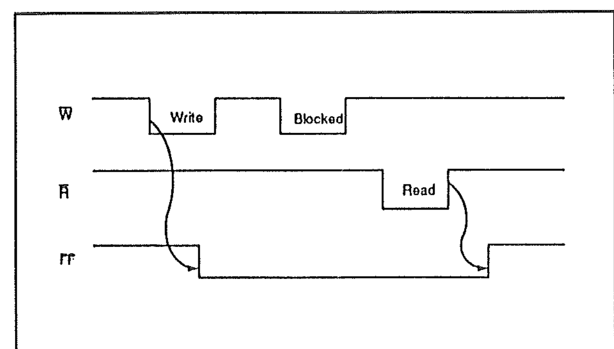
Figuur 3/6.13.6-7: Functioneel blokschema van een IDT7201/02 FIFO.

Timing van de FIFO

Na resetten wordt de leeg-vlag $\overline{EF}$ LAAG en zodra data wordt ingeschreven weer HOOG. $\overline{EF}$ gaat dan pas weer LAAG als alle informatie is uitgelezen (figuur 3/6.13.6-8). Als het aantal opgenomen data-elementen de helft van het totaal aantal plaatsen bereikt, gaat de halfvolvlag $\overline{HF}$ LAAG. Zijn precies alle plaatsen in de FIFO van data voorzien, dan gaat de vol-vlag ($\overline{FF}$) LAAG om aan te geven dat er geen plaats meer is (figuur 3/6.13.6-9).

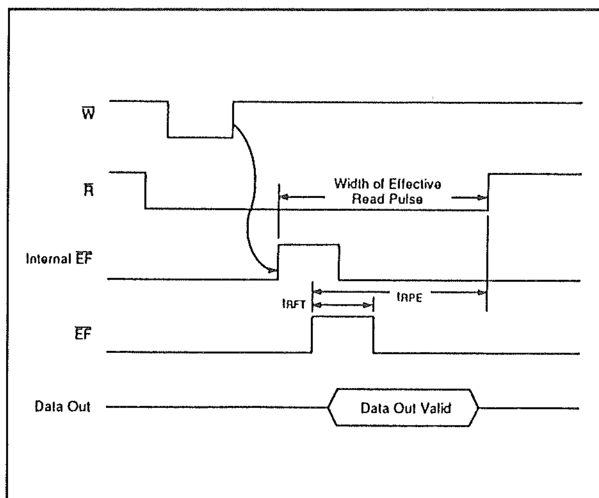


Figuur 3/6.13.6-8: Wanneer de FIFO slechts één woord bevat gaat de leeg-vlag ($\overline{EF}$) op de dalende flank van het leessignaal ($\overline{R}$) LAAG. Alle volgende leespulsen worden na het HOOG gaan van $\overline{R}$ geblokkeerd, terwijl $\overline{EF}$ LAAG blijft. Op de stijgende flank van de schrijfpuls $\overline{W}$ vervalt de leeg-conditie.



Figuur 3/6.13.6-9: Wanneer de FIFO op één woord na vol is, zal $\overline{FF}$ op de dalende flank van $\overline{W}$ LAAG gaan. Alle volgende schrijfpulsen worden na het HOOG gaan van $\overline{W}$ genegeerd, terwijl $\overline{FF}$ LAAG blijft. Op de stijgende flank van $\overline{R}$ wordt de vol-conditie opgeheven.

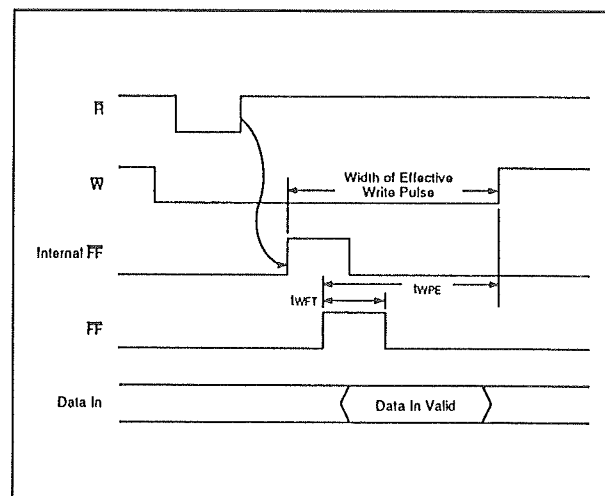
Wanneer de FIFO leeg is en $\overline{R}$ LAAG wordt gehouden voordat er schrijfpulsen LAAG gaan, vindt er automatische "Leesdoorstroming" (Read data flow-through) plaats. De stijgende flank van $\overline{W}$ maakt $\overline{EF}$ HOOG, maar omdat $\overline{R}$ LAAG wordt gehouden treedt hierdoor een leescyclus op. Door dit lezen wordt $\overline{EF}$ weer LAAG (figuur 3/6.13.6-10). Op $\overline{EF}$ zullen zodoende HOOG gaande pulsen optreden met een breedte van minstens t_{RFT}.
Op dezelfde wijze kan een "Schrijfdoorstroming" (Write data flow-through) optreden als de FIFO vol is en $\overline{W}$ LAAG wordt gehouden voordat een leespuls LAAG gaat. De stijgende flank van $\overline{R}$ maakt $\overline{FF}$ HOOG, maar omdat $\overline{W}$ LAAG wordt gehouden treedt hierdoor een schrijfcyclus op. Dit schrijven maakt dat $\overline{FF}$ weer LAAG gaat (figuur 3/6.13.6-11). Op $\overline{FF}$ zullen dus pulsen HOOG gaan met een breedte van minstens t_{WFT}.



Figuur 3/6.13.6-10: Bij automatische lees-doorstroming wordt $\overline{R}$ LAAG gehouden.

Bredere FIFO's

Wanneer in een toepassing FIFO's moeten worden gebruikt die breder zijn dan bestaande (of in voorraad gehouden) exemplaren, kunnen zij gemakkelijk worden verbreed door ze parallel te schakelen. In figuur 3/6.13.6-12 is een voorbeeld getekend van een FIFO voor 18 bit brede woorden. De status-vlaggen kunnen willekeurig van één van beide FIFO's worden gedetecteerd. Bij oudere typen kan externe logika nodig zijn om combinaties van Input-Ready en Output-Ready signalen te gebruiken (figuur 3/6.13.6-13).

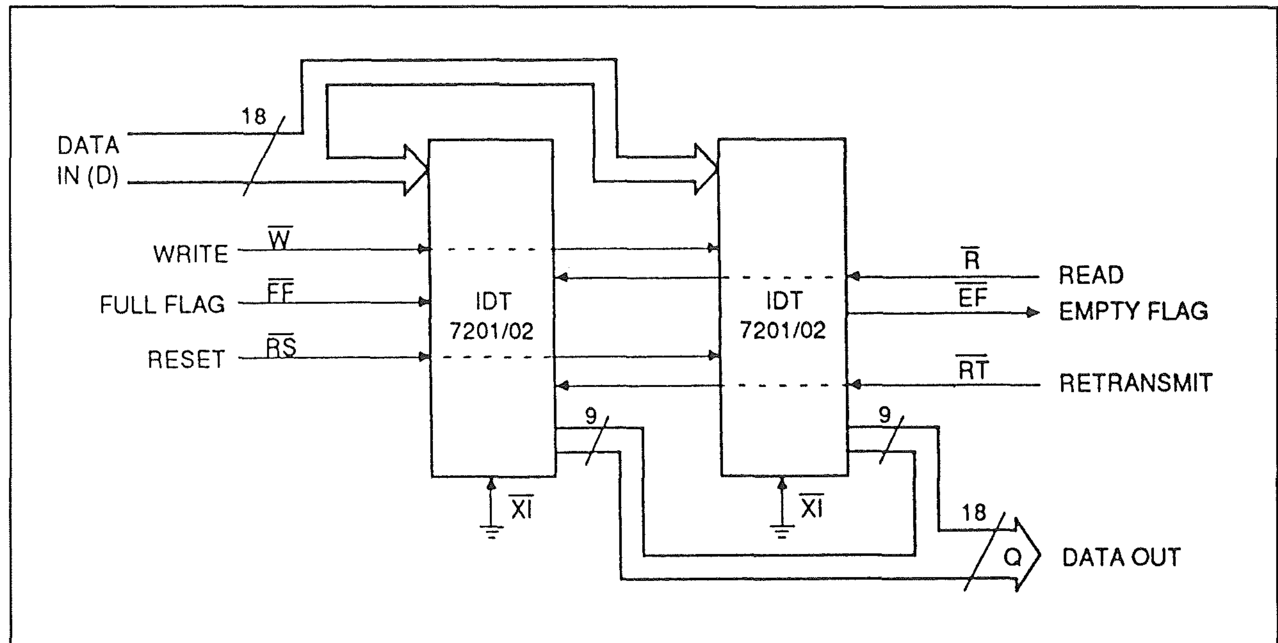


Figuur 3/6.13.6-11: Automatische schrijf-doorstroming treedt op als $\overline{W}$ LAAG wordt gehouden.

Langere FIFO's

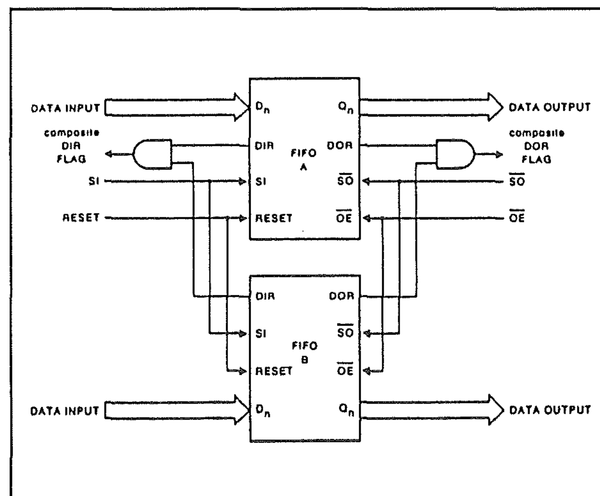
Figuur 3/6.13.6-14 toont hoe eenvoudig de lengte (aantal woorden) bij sommige typen vergroot kan worden. De werking is als volgt. Data die met SIA door FIFO A wordt opgenomen komt na een vertraging aan de uitgang daarvan. Omdat $\overline{SOA}$ HOOG is wordt een DORA puls gegenereerd (= SIB). Na nog een vertraging komt de data op de uitgang van FIFO B. Dit gaat door totdat FIFO B vol is. DIRB (= $\overline{SOA}$) gaat LAAG, zodat geen SIB-pulsen meer worden opgewekt. De rest van de data komt verder alleen in FIFO A terecht. Bij uitlezen wordt SOB via DIRB aan FIFO A doorgegeven als SOA.

6.13 Digitale geheugens



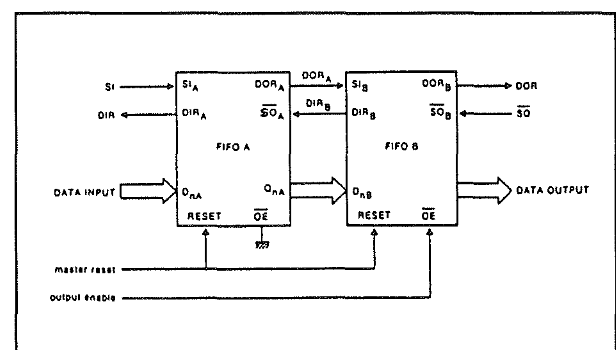
Figuur 3/6.13.6-12: Uitbreiding van twee 7201/02-typen tot een FIFO met 18-bits woordbreedte.

Alle data schuift dus één plaats naar rechts en DIRA wordt HOOG.



Figuur 3/6.13.6-13: Bij sommige typen is voor grotere woordbreedten een combinatie van de DIR- en DOR-vlaggen nodig.

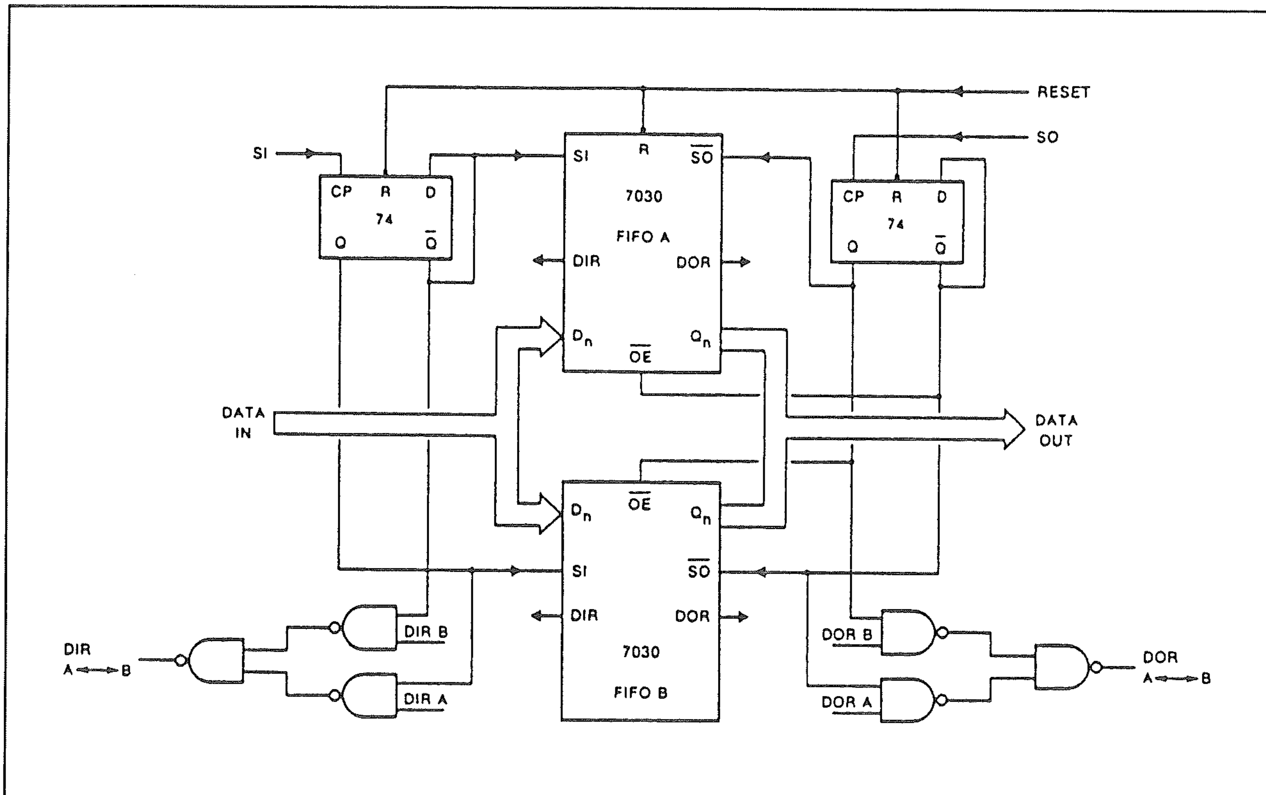
Als FIFO A toevallig een langzaam type is en FIFO B een zeer snel type, kan het voorkomen dat de DIRB-puls te kort is om als $\overline{SO}$A-puls te dienen, zodat FIFO A niet goed werkt. Omgekeerd kan het ook gebeuren dat DORA te kort duurt om als SIB te werken (als FIFO B langzamer is dan FIFO A). In deze gevallen moeten de pulsen naar de andere FIFO worden verlengd.



Figuur 3/6.13.6-14: Vergroting van het aantal woorden door in serie schakelen van twee (of meer) FIFO's.

Bij de schakeling zoals in figuur 3/6.13.6-14 neemt de doorvaltijd toe met het aantal FIFO's dat in serie wordt geschakeld.

6.13 Digitale geheugens

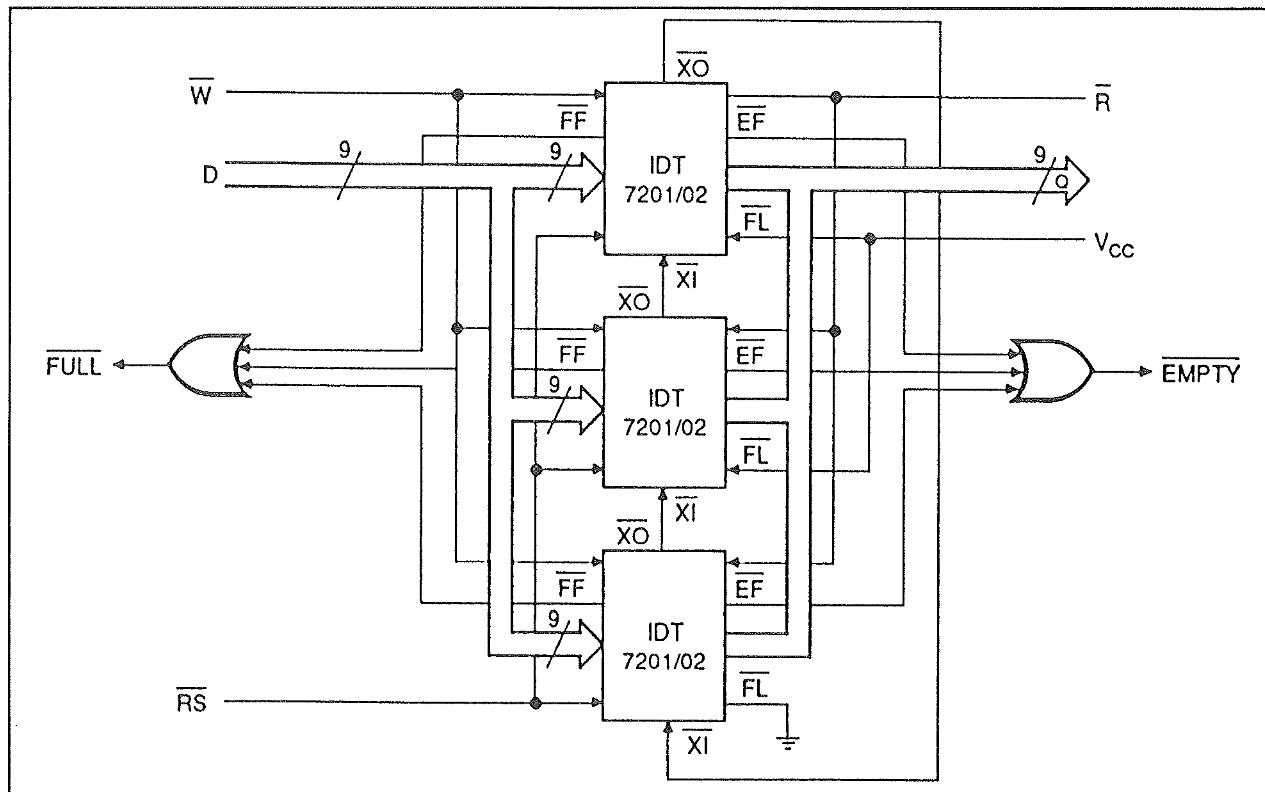


Figuur 3/6.13.6-15: Verdubbeling van het aantal woorden door middel van gemeenschappelijke data-in en data-uit bussen. Hierbij neemt de doorvaltijd niet toe.

Het spreekt vanzelf dat bij zeer lange FIFO's de doorvaltijd dan ontoelaatbaar lang dreigt te worden. Een oplossing hiervoor is de FIFO's parallel te schakelen, zoals figuur 3/6.13.6-15 laat zien. De flipflop's zorgen er dan voor dat beide FIFO's om de beurt worden gebruikt. Bij langere FIFO's moet natuurlijk een uitgebreider decodering worden toegepast.

Met twee-pointer FIFO's zoals de 7201 en 7202 kunnen alle data-ingangen met elkaar verbonden worden, evenals de data-uitgangen (zie figuur 3/6.13.6-16). Hierdoor ontstaat een parallelle architectuur die ook wordt gebruikt bij gewone RAM's om diepere geheugens te verkrijgen. Omdat FIFO's geen chip-selects en externe decoderingen kennen moet het kiezen van de juiste FIFO intern gebeuren. In de 7201/02 wordt dit bereikt door een unieke seriële structuur. De eerste (of master) FIFO wordt geïdentificeerd door de $\overline{FL}$-ingang te aarden. Van alle overige FIFO's in de schakeling moet de $\overline{FL}$-ingang worden opgetrokken naar V_{CC}. De $\overline{XO}$-uitgang van de eerste FIFO wordt verbonden met de $\overline{XI}$-ingang van de volgende FIFO in de rij, enzovoorts. De $\overline{XO}$-uitgang van de laatste FIFO wordt tenslotte aangesloten op de $\overline{XI}$-ingang van de eerste. Na het resetten staan de aktieve lees- en schrijfpointers R en W op de eerste FIFO. Als de schrijf-pointer (W) aan het einde van de eerste FIFO is gekomen ontstaat een puls op de $\overline{XO}$-uitgang, waardoor de schrijfpointer aan het begin van de tweede FIFO wordt geaktiveerd, terwijl tegelijkertijd de schrijf-pointer van de eerste wordt uitgeschakeld.

6.13 Digitale geheugens



Figuur 3/6.13.6-16: Opbouw van een snelle, langere FIFO waarbij de $\overline{XI}$- en $\overline{XO}$-lijnen worden gebruikt. De data-ingangen zijn met elkaar verbonden, evenals de data-uitgangen.

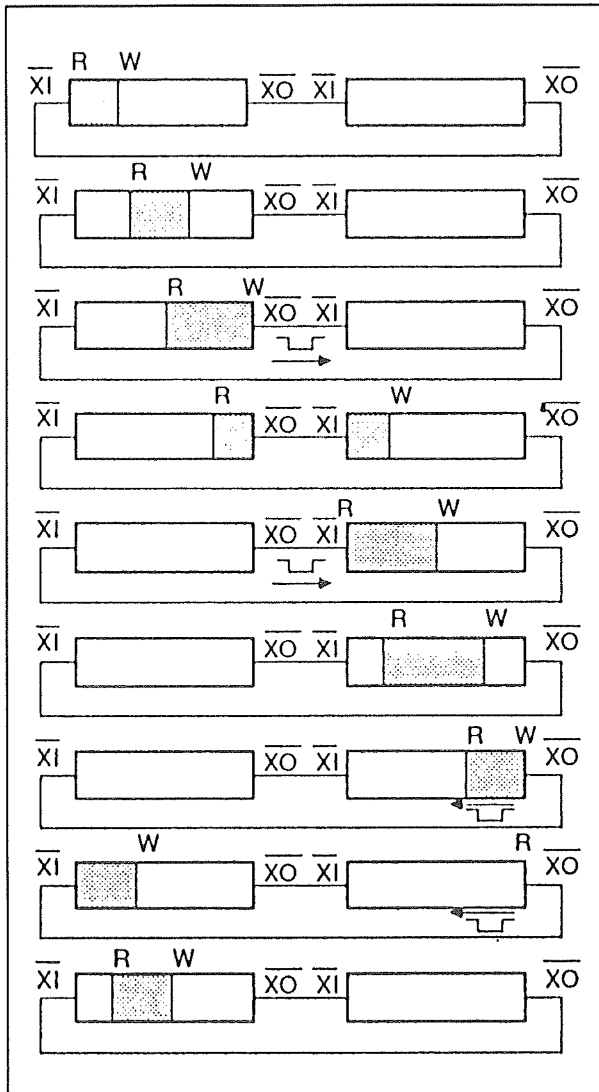
De besturing van Write-Enable wordt zodoende doorgegeven naar de tweede FIFO. Wanneer de aktieve lees-pointer (R) het einde van de eerste FIFO bereikt aktiveert hij via een tweede puls op de $\overline{XO}$-uitgang de lees-pointer in de tweede FIFO en schakelt hij zichzelf uit. In figuur 3/6.13.6-17 is de verplaatsing van de pointers met behulp van $\overline{XO}$ en $\overline{XI}$ over twee FIFO's te zien. In deze ringstructuur loopt de lees-pointer altijd na op de schrijfpointer (hij kan deze niet passeren).

In figuur 3/6.13.6-18 wordt tenslotte getoond hoe de positie van de pointers is wanneer een grotere hoeveelheid data wordt opgenomen. Alleen na een reset bevinden beide pointers zich voorin de eerste FIFO. Nadat alle data is uitgelezen staan W en R weer tegenover elkaar, maar dan op een willekeurige plaats.

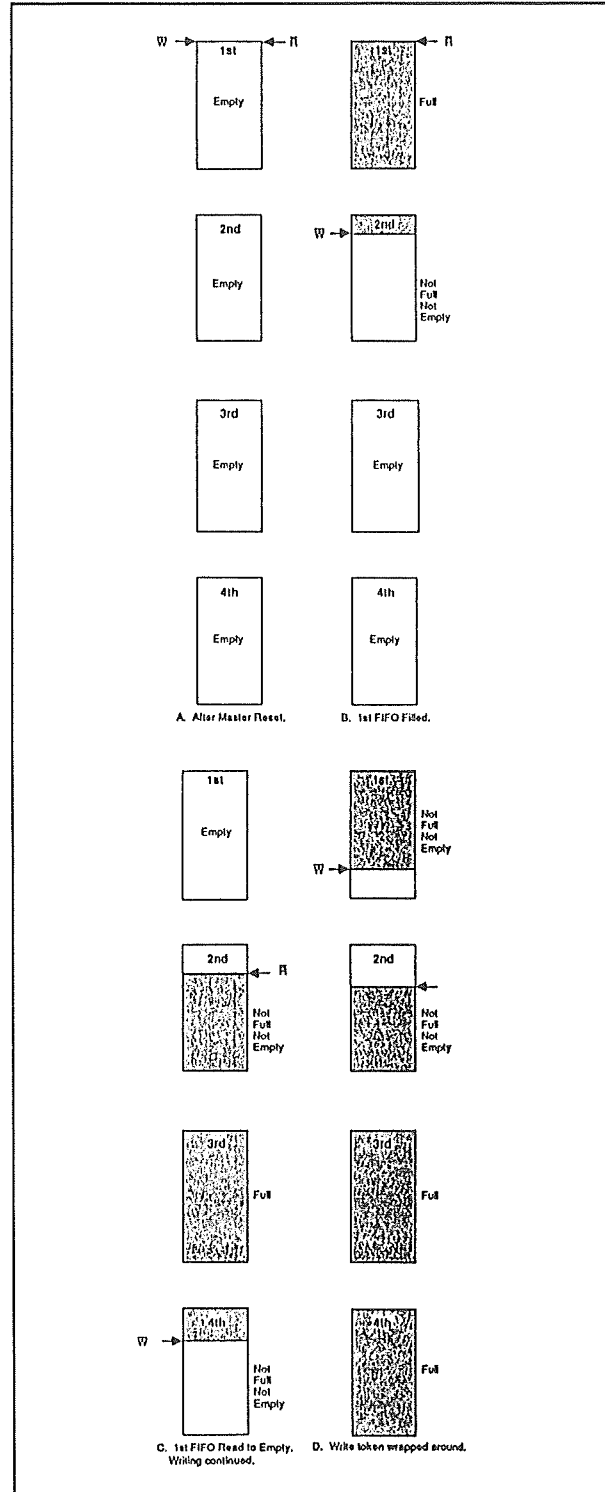
De werking is als volgt:

- A: na een master-reset staan W en R aan het begin van FIFO 1;
- B: FIFO 1 is gevuld en resterende data komt in FIFO 2 terecht;
- C: FIFO 1 is leeggelezen; het schrijven gaat door;
- D: de leeggelezen FIFO 1 wordt ook beschreven (wrapped around).

6.13 Digitale geheugens



Figuur 3/6.13.6-17: De verplaatsing van de lees- en schrijfpointers over twee FIFO's. Telkens wanneer W of R de grens van een FIFO overschrijdt wordt een $\overline{XO}$-puls gegenereerd.



Figuur 3/6.13.6-18: Positie van data en pointers in vier FIFO's die op de manier van figuur 3/6.13.6-16 zijn geschakeld.

3/6.14

Binaire multipliers

Inleiding
De binaire multipliers kunnen worden opgesplitst in twee groepen:
- de binaire rate multipliers;
- de "echte" binaire multipliers.

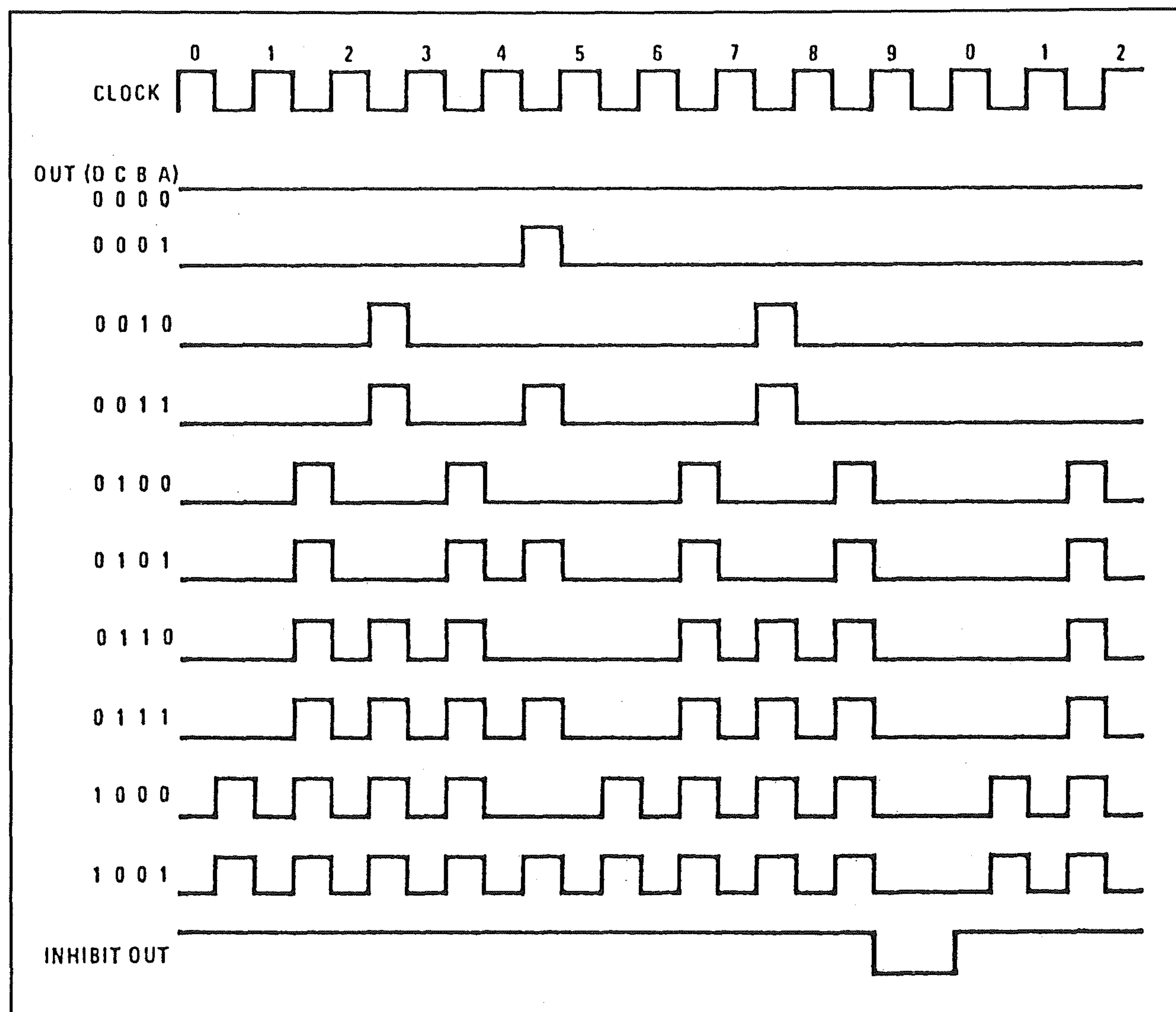
Rate multipliers
Dit zijn eigenlijk delers waarbij de frequentie van een clocksignaal door een instelbaar getal wordt gedeeld.
Als voorbeeld wordt een BCD Rate Multiplier van het type 4527 besproken. In het tijddiagram van figuur 3/6.14-1 ziet men dat van elke 10 clockpulsen er n worden doorgegeven aan de uitgang.
Hierbij is n het ingangsgetal. Is bijvoorbeeld n = 6 (BCD 0110), dan komen er per 10 clockpulsen 6 aan de uitgang.

Rate Multipliers kunnen worden toegepast voor rekenkundige bewerkingen, zoals vermenigvuldigen en delen, bij analoog/digitaal- en digitaal/analoogomzetting en als frequentiedeler. Bij het laatste moet rekening worden gehouden met het feit dat de uitgangspulsen meestal niet gelijkmatig over de tijd zijn verdeeld.

Binaire multipliers
De werking hiervan is wat ingewikkelder. Vandaar wordt eerst het binaire vermenigvuldigen behandeld. De hierbij geldende regels zijn geschetst in figuur 3/6.14-2. Wanneer men zich daaraan houdt is er eigenlijk geen verschil tussen binair en decimaal vermenigvuldigen. In het voorbeeld van figuur 3/6.14-3 is te zien dat de methode identiek is. Een binair vermenigvuldigtal 1101 (= decimaal 13) wordt vermenigvuldigd met de vermenigvuldiger 1010 (= decimaal 10). Eerst vindt vermenigvuldiging plaats van de laagste bit (1-en) van de vermenigvuldiger en het vermenigvuldigtal. Het resultaat daarvan is 0000. Daarna is het de beurt van het 2-bit van de vermenigvuldiger, met als tussenresultaat 1101. Merk op dat dit tussenprodukt één plaats naar links is verschoven ten opzichte van het eerste tussenprodukt.
Op deze wijze wordt de gehele vermenigvuldiging uitgevoerd tot het eindprodukt 100000110 is bereikt (= decimaal 130).

In figuur 3/6.14-4 is de gehele procedure nogmaals te zien, maar dan aangevuld met het proces dat door de binaire logika wordt gebruikt bij de "tel-op-en-schuif" methode (add-and-shift). De regels 1 en 2 bevatten de beide te vermenigvuldigen binaire getallen. Op regel 3 ziet men dat de 1-en vermenigvuldiger (die gelijk is aan 0) wordt vermenigvuldigd met het vermenigvuldigtal. Het eerste partiële produkt is 0000. Regel 4 toont een "schuif-links" voor het volgende partiële produkt (of een "schuif-rechts" voor het eerste partiële produkt).

6.14 Binaire multipliers



Figuur 3/6.14-1: Tijddiagram van een BCD Rate Multiplier.

0	0	1	1
× 0	× 1	× 0	× 1
0	0	0	1

Figuur 3/6.14-2: Regels voor binair vermenigvuldigen.

Decimal	Binary	
13	1101	Multiplicand
× 10	× 1010	Multiplier
00	0000	1st partial product
13	1101	2nd partial product
130	0000	3rd partial product
	1101	4th partial product
	10000010	Product

Figuur 3/6.14-3: Een voorbeeld van binair vermenigvuldigen.

6.14 Binaire multipliers

1	Multiplicand	1101	
2	Multiplier	× 1010	
3	1st partial product	0000	1s multiplier bit = 0. Write 0000.
4		----0	Shift left.
5	2nd partial product	11010	2s multiplier bit = 1. Copy multiplicand: 1101.
6		11010	Add 1st and 2nd partial products.
7		----00	Shift left.
8	3rd partial product	000000	4s multiplier bit = 0. Write 0000.
9		011010	Add 1st and 2nd plus 3rd partial products.
10		----000	Shift left.
11	4th partial product	1101000	8s multiplier bit = 1. Copy multiplicand: 1101.
12	Product	10000010	Add 1st, 2nd, and 3rd plus 4th partial products.

Figuur 3/6.14-4: Hetzelfde voorbeeld als in figuur 3/6.14-3, aangevuld met de procedure die door logische schakelingen wordt gebruikt voor binaire vermenigvuldiging.

Op regel 5 wordt de 2-en bit (=1) vermenigvuldigd met het vermenigvuldigtal. Dit levert 1101 op en wordt aangevuld met de meest rechtse 0.
Het tweede partiële produkt bedraagt dus 11010.
Op regel 6 wordt het eerste partiële produkt bij het tweede opgeteld met 11010 als resultaat.
Regel 7 toont nogmaals een "schuif-links". De 4-en bit van de vermenigvuldiger is gelijk aan 0, zodat het derde partiële produkt 0000 wordt. Op regel 8 wordt hier 00 aan toegevoegd en wordt het volledige derde partiële produkt 00000. Op regel 9 worden de regels 6 en 8 bij elkaar opgeteld, enzovoorts.

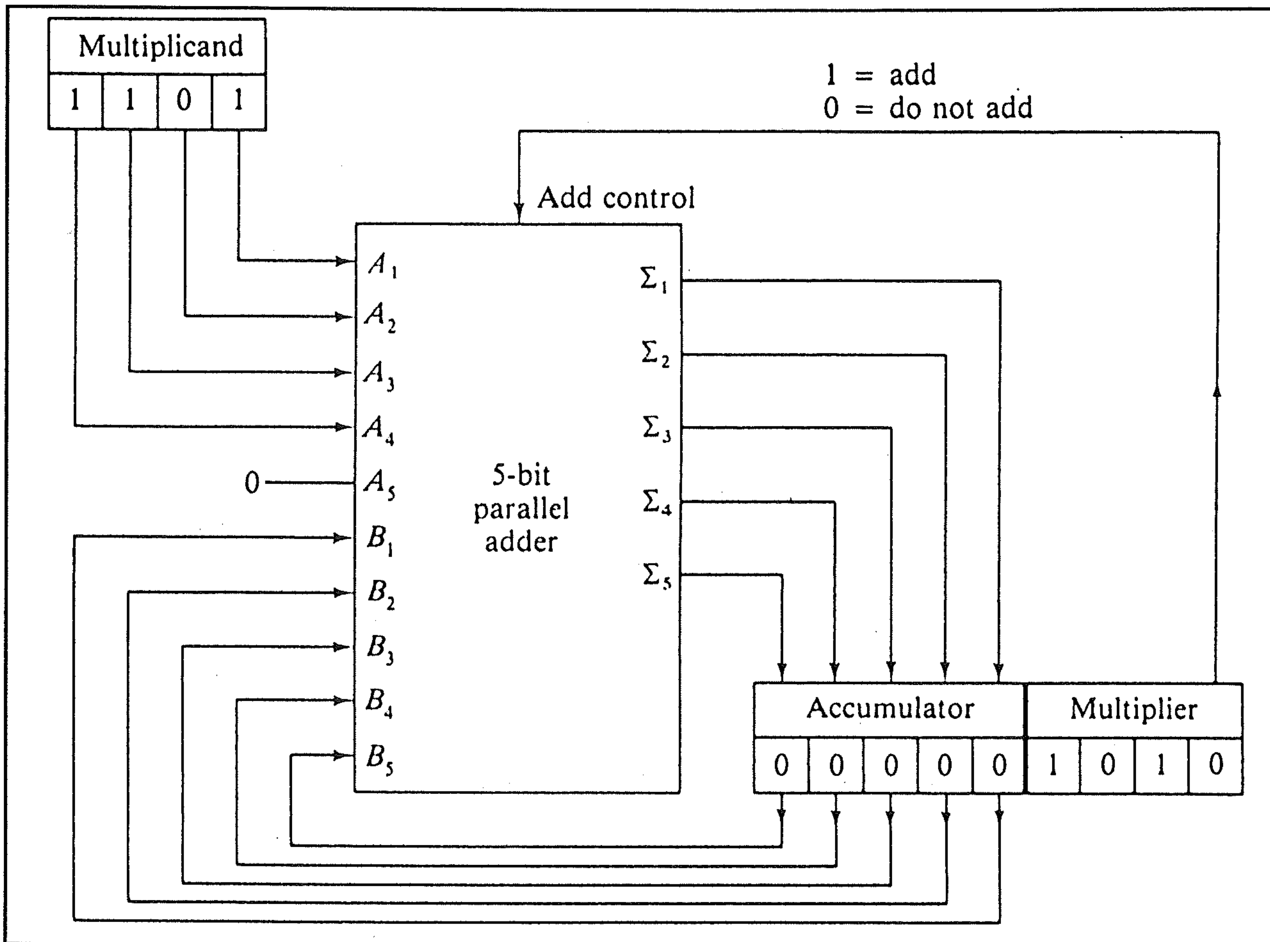
Deze "tel-op-en-schuif methode" kan met digitale schakelingen worden uitgevoerd.

Wanneer men het binaire vermenigvuldigingsproces nauwkeurig bekijkt, dan vallen drie belangrijke dingen op:
- Partiële produkten zijn 0000 als de vermenigvuldiger-bit gelijk aan 0 is, of gelijk aan het vermenigvuldigtal als de vermenigvuldiger-bit gelijk aan 1 is.
- Het eindresultaat kan tweemaal zo lang zijn als het vermenigvuldigtal.
- Bij het optellen wordt het eerste partiële produkt één plaats naar rechts geschoven ten opzichte van het tweede partiële produkt.

Binaire multiplier
Deze feiten vormen de basis van een binaire vermenigvuldiger zoals geschetst is in figuur 3/6.14-5. Deze schakeling bevat een 5-bit parallelle opteller (adder) met een add/do-not-add besturing (wel/niet optellen).
Het register dat het vermenigvuldigtal (multiplicand) bevat is een 4-bit schuifregister (links boven). Rechts onder bevinden zich een 5-bit accumulator-register en een 4-bit vermenigvuldiger-register (multiplier).

Net als in het voorbeeld van figuur 3/6.14-4 wordt nu de multiplicand geladen met 1101 en de multiplier met 1010. De accumulator wordt leeggemaakt (00000). Deze situatie komt overeen met stap 1 in figuur 3/6.14-6. Stap 2 is het optellen uit de optel-en-schuif procedure.

6.14 Binaire multipliers

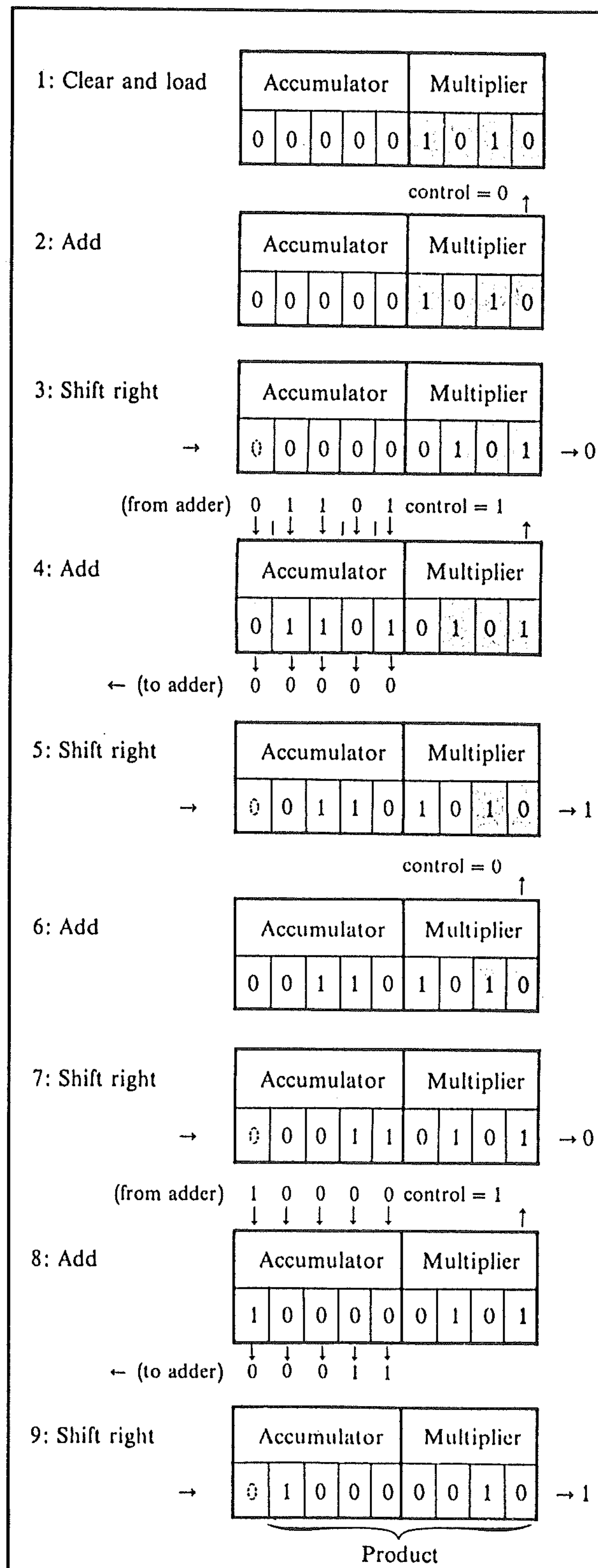


Figuur 3/6.14-5: Een binaire vermenigvuldiger.

De minst belangrijke bit (LSB) van de multiplier wordt toegevoerd aan de add-besturing van de 5-bit adder. In dit geval is het een 0, zodat geen optelling wordt uitgevoerd.

Alle registers blijven gelijk. Deze stap komt overeen met regel 3 in figuur 3/6.14-4. Het eerste partiële produkt (0000) komt in het accumulator-register. Bij stap 3 worden de inhouden van de accumulator en de multiplier één plaats naar rechts geschoven. Hierbij wordt van links een 0 ingebracht en gaat aan de rechterzijde een 0 verloren. Zoals aan het donkere gedeelte te zien is, blijven slechts drie bits van de originele waarde in de multiplier over. Bij stap 4 wordt weer opgeteld. De LSB in het multiplier-register laat de 5-bit adder optellen. De 00000 uit de accumulator wordt nu bij de 01101 uit het multiplicand-register opgeteld. Het resultaat (01101) wordt opgeborgen in de accumulator. De som van de partiële produkten op dit moment (011010) is te zien in het lichte gedeelte van zowel de accumulator als de multiplier. Dit komt overeen met regel 6 van figuur 3/6.14-4. Bij stap 5 wordt weer naar rechts geschoven. Van links komt een 0 binnen en naar rechts gaat nu een 1 verloren. Bij stap 6 gaat het optellen nu niet door, omdat de LSB van de multiplier nu een 0 is en blijven de registers onveranderd.

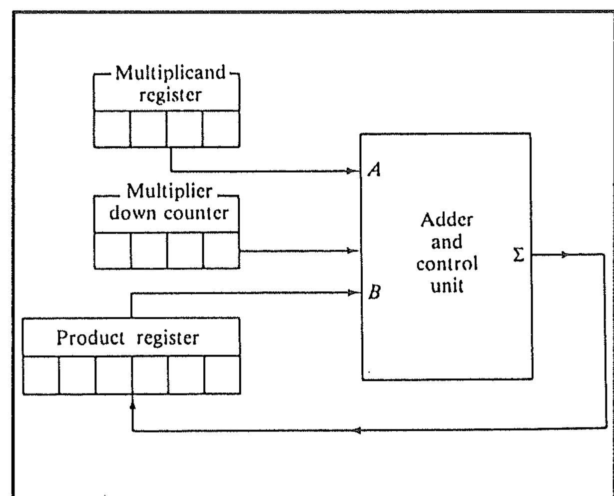
6.14 Binaire multipliers



Figuur 3/6.14-6: Werking van de binaire vermenigvuldiger.

Bij stap 7 wordt nogmaals naar rechts geschoven, waardoor van links een 0 binnenkomt en naar rechts een 0 verloren gaat. Bij stap 8 wordt nu wel opgeteld omdat de LSB een 1 is. De inhoud van de accumulator (00011) wordt op de B-zijde van de adder gezet en opgeteld bij de inhoud van de multiplicand die onveranderd 1101 blijft. De som (10000) wordt in de accumulator geladen en bij stap 9 wordt voor de laatste maal naar rechts geschoven. Het eindresultaat bevindt zich nu in de accumulator- plus de multiplier-registers.

Vermenigvuldigen met herhaald optellen
De tel-op-en-schuif procedure kan met logische schakelingen worden gebouwd of als computerprogramma worden geschreven. Een andere methode van vermenigvuldigen is natuurlijk het herhaald optellen.
Wanneer bijvoorbeeld 6 x 5 moet worden uitgerekend, kan 6 + 6 + 6 + 6 + 6 (= 30) worden uitgevoerd.
Het spreekt vanzelf dat één van beide getallen dan in een teller wordt geladen die na elke optelling met één wordt verlaagd (zie figuur 3/6.14-7).



Figuur 3/6.14-7: Vermenigvuldigen door middel van herhaald optellen.

6.14 Binaire multipliers

3/6.15

Werking en principes van flash-geheugens

Inleiding

De voorgeschiedenis

Vele elektronische apparaten hebben bij het inschakelen een stuk programmacode nodig, waarmee de processor in het apparaat kan worden geïnitialiseerd en waarmee het bedrijfssysteem kan worden geladen. Tot voor kort kon deze programmacode alleen in een ROM-, EPROM- of EEPROM-geheugen worden opgeslagen. Dat waren namelijk de enige geheugentypen waarin men eenmalig code kon opslaan die ook na het uitschakelen van de voedingsspanning in het geheugen bleef staan.

ROM-geheugens hebben als nadeel dat de productie ervan tamelijk duur is, zeker als men maar weinig exemplaren nodig heeft.
EPROM's hebben als voordeel dat zij zélf te programmeren zijn met de geschikte apparatuur en dat de code nadien ook weer gewist kan worden. Voor dat wissen is echter een tamelijk uitgebreide procedure noodzakelijk, waarbij de chip van de EPROM een bepaalde tijd wordt blootgesteld aan een forse dosis ultraviolette straling. Nadien kan men weer met een EPROM-programmer een nieuwe code in het geheugen inlezen. Het zou echter zeer handig zijn als er een permanent geheugen beschikbaar was, waarin men op een even eenvoudige manier nieuwe code zou kunnen schrijven als dat het geval is bij RAM-geheugens. En natuurlijk zou, de naam "permanent" zegt het al, deze code bij het wegvallen van de voedingsspanning in de geheugencellen bewaard moeten blijven. Kortom, de ideale permanente geheugen-chip zou de voordelen van een traditionele EPROM-chip moeten combineren met de voordelen van een RAM-chip.

De flash-technologie

Dergelijke geheugens zijn er nu! Deze werken volgens een speciale technologie en worden aangeduid met de verzamelnaam "flash-geheugens". In dit hoofdstuk zal de principiële werking van deze technologie worden besproken en zal als praktisch voorbeeld één flash-geheugen, namelijk de 28F256, uitvoerig behandeld worden.

Een Toshiba/Intel ontwikkeling

Reeds in 1983 werd begonnen met onderzoek naar halfgeleidergeheugens waarbij de informatie niet alleen kon worden vastgehouden bij het uitschakelen van de voedingsspanning, maar ook veranderd. In 1985 werd door Toshiba de flash-geheugen-technologie geïntroduceerd, die in 1988 als gevolg van onderzoek door Intel op het gebied van OTP's (One-Time Pro-

grammable EPROM's) op grote schaal doorbrak. Bij flash-geheugens wordt de geprogrammeerde informatie vastgehouden (ook bij weghalen van de voedingsspanning), maar vervalt het langdurige wissen met ultra-violet licht dat bij EPROM's nodig is. Flash-EPROM's kunnen in een PROM-programmeervoet of op de printkaart zélf in de definitieve schakeling worden geprogrammeerd. Men heeft dus de mogelijkheid van "On-Board Programmeren", een nieuw begrip dat tegenwoordig met het letterwoord "OBP" door het leven gaat.

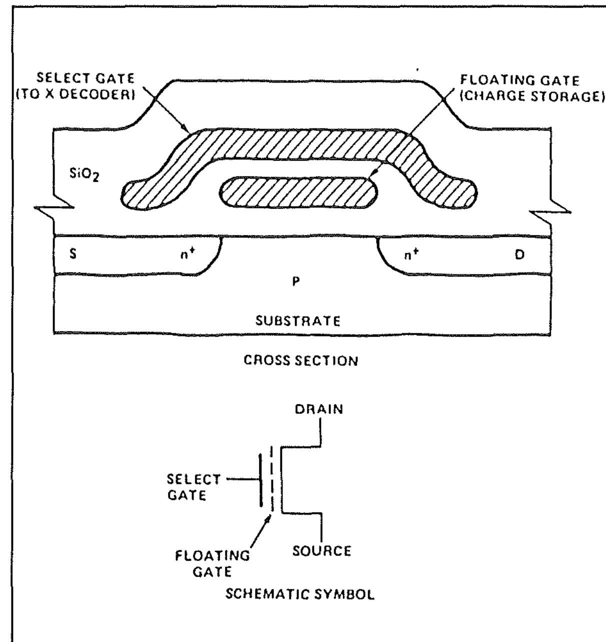
Soorten flash-geheugens

Wat opbouw betreft kunnen de flash-geheugens in twee hoofdgroepen worden onderverdeeld. De eerste groep is gebaseerd op de EPROM en heeft één transistor per geheugencel en de tweede groep berust op het EEPROM-principe en heeft twee transistoren per cel.
Men moet dus onderscheid maken tussen flash-EPROM's en flash-EEPROM's.

Begin 1992 heeft AMD echter de ontwikkeling van een nieuw type, de "5 V-only negative gate erase Flash-EPROM", aangekondigd. Dit flash-geheugen bevat ook ééntransistor geheugencellen, maar werkt op een enkele voedingsspanning van 5 V. Inwendig wordt door middel van een "charge-pump" een negatieve wis- en programmeerspanning van -10,5 V opgewekt. Om compatibel te zijn met de 5 V/12 V-typen is de V_{pp}-pen hiervan open gelaten.

Werking van "normale" EPROM's

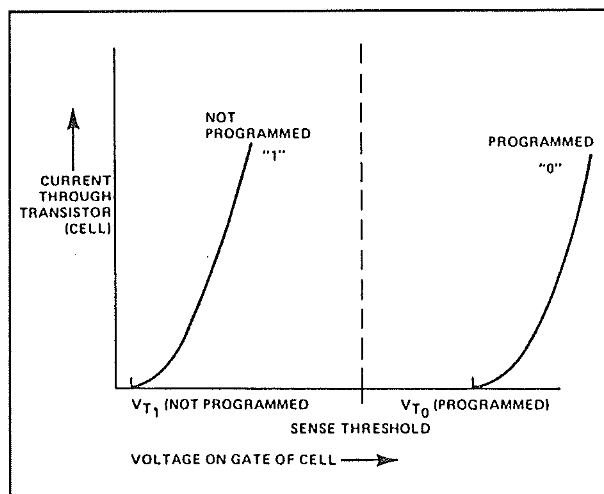
Alvorens de werking van flash-EPROM's te bespreken is het nuttig even te recapituleren hoe "normale" EPROM's werken. Een EPROM geheugencel is gebaseerd op een zogenaamde zwevende gate in een NMOS-transistor. De fysische opbouw van een dergelijke transistor is getekend in figuur 3/6.15-1.



Figuur 3/6.15-1: De samenstelling van een EPROM geheugencel.

Om een dergelijke zwevende gate op te laden moeten zowel de select-gate als de drain op een voldoende hoge positieve spanning worden gebracht. De source en het substraat moeten op massa-potentiaal staan. Door het sterke elektrische veld dat dan ontstaat worden elektronen in het pinch-off gebied van de transistor versneld. Sommige van deze elektronen krijgen voldoende energie om in de geleidingsband van het basismateriaal SiO_2 te geraken. Bij normale spanningen wordt de zwevende gate hierdoor geïsoleerd. Deze elektronen worden aangetrokken door het positieve potentiaal van de zwevende gate. De aanwezigheid van lading op de zwevende gate veroorzaakt een verschuiving van de gate-drempelspanning, zoals getekend in figuur 3/6.15-2.

6.15 Werking en principes van flash-geheugens



Figuur 3/6.15-2: Het verschuiven van de drempelspanning door het programmeren van de geheugencel.

In de oorspronkelijke toestand heeft de cel een zeer lage drempel, zodat de transistor gaat geleiden bij selectie van de cel door middel van de select gate. Door het programmeren wordt de drempel naar een hogere spanning verschoven, waardoor de transistor bij selectie niet meer kan geleiden. Indien een "H" in de cel is geprogrammeerd zal door de selectie een hogere stroom van source naar drain gaan vloeien dan wanneer een "L" in de cel was geladen.

De lading die zich op de zwevende gate bevindt kan er door bestralen met ultraviolet licht weer afgehaald worden. Dat is de reden waarom EPROM's zijn voorzien van een kwarts venstertje. Er bestaat een bepaalde samenhang tussen het vermogen van de UV-lichtbron, de afstand tot de EPROM en de voor het wissen benodigde tijd. Voor een UV-lamp die bijvoorbeeld 12 mW/cm^2 afgeeft en die op 2 à 3 cm van de chip is geplaatst bedraagt de wistijd ongeveer 15 tot 20 minuten.

Werking van de flash-EPROM

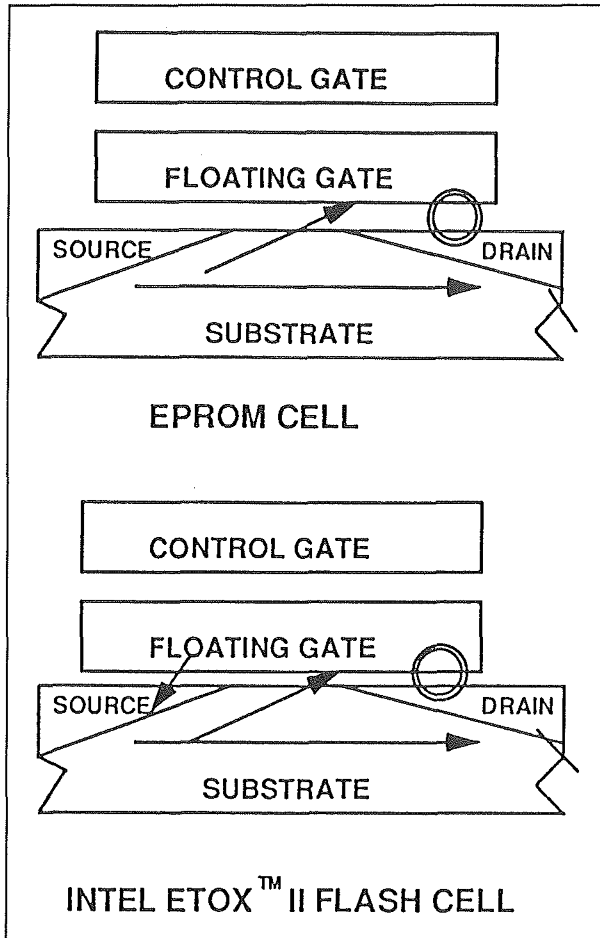
Bij de oudste serie (Flash-I) werd alle informatie tegelijk elektrisch gewist. Bij Flash-II kan de opgeslagen informatie selectief (per blok) worden gewist en herschreven. Flash-II geheugencellen zijn net als EPROM's opgebouwd uit enkele transistoren met een zwevende gate waarop lading kan worden opgeslagen. De principiële samenstelling van een normale EPROM-cel en een flash-cel wordt vergeleken in figuur 3/6.15-3. Door de toegepaste ETOX-technologie kan de oxydelaag ongeveer 20 maal dunner zijn dan bij EPROM's, waardoor degeneratie wordt voorkomen.

Dergelijke flash-geheugens kunnen ruim 100.000 maal worden geprogrammeerd door met een hulpspanning van 12 V op de gate een "hot electron injection" mechanisme te aktiveren. Bij het wissen wordt deze spanning ook gebruikt (maar dan op de source) om de opgeslagen lading van de zwevende gate naar de source te laten afvloeien. Dit wordt toegelicht in figuur 3/6.15-4.
Bij Flash-II geheugens worden dus twee voedingsspanningen gebruikt: +5 V en +12 V, hetgeen ook de veiligheid ten goede komt. Dit soort geheugens wordt onder andere gefabriceerd door Intel, AMD, Mitsubishi, Hitachi en Catalyst.
Ter vergelijking is in figuur 3/6.15-5 geschetst hoe het wissen van een "5 V/negatieve gate flash-EPROM" gebeurt. De negatieve spanning wordt met behulp van een inwendige ladingspomp (charge pump) opgewekt.

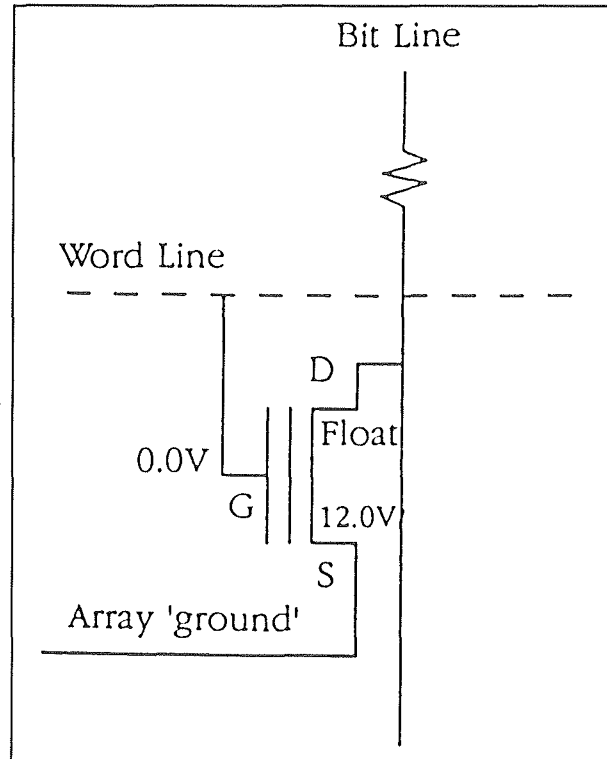
Flash-EEPROM's

Er zijn ook op de EEPROM-technologie gebaseerde flash-geheugens die met slechts één voedingsspanning werken.

6.15 Werking en principes van flash-geheugens



Figuur 3/6.15-3: Vergelijking van de samenstelling van een "gewone" EPROM-cel (boven) en een flash-cel (onder).



Figuur 3/6.15-4: Het wissen van 5 V/12 V flash EPROM's.

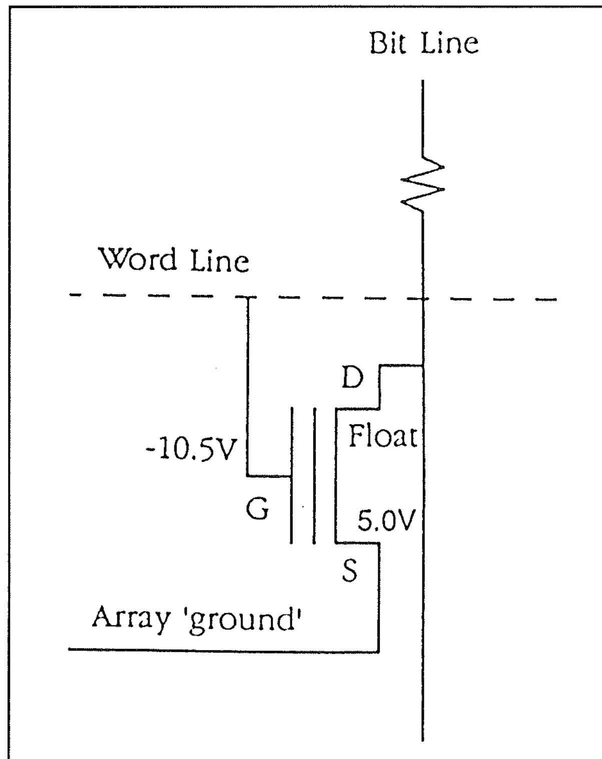
De geheugencellen hiervan bestaan echter telkens uit twee transistoren, terwijl inwendig extra schakelingen (timers, ladingspompen voor de interne opwekking van de benodigde hulpspanning en "write-protect" schakelingen) nodig zijn. Deze geheugens hebben het voordeel dat de informatie per byte kan worden gewist. Daar staat tegenover dat de dichtheid, het aantal geheugencellen per oppervlakte-eenheid, geringer is en deze daardoor dus duurder zijn. Fabrikanten van deze typen zijn onder andere Texas Instruments, Catalyst, Atmel en Toshiba.

Toepassingen

Er zijn twee belangrijke toepassingsgebieden voor flash-geheugens.

- Embedded systemen, door microcontroller bestuurd
  Onder embedded systemen vallen bijvoorbeeld disk-controllers, robots, testapparatuur, cellulaire telefoons (code), medische toepassingen, consumenten-elektronica (pay-TV) en auto-elektronica.
- Elektronische gegevensverwerking
  Onder gegevensverwerking vallen bijvoorbeeld desktop, laptop, notebook en palmtop PC's (voor de BIOS en in plaats van diskettes), werkstations en terminals (beveiliging), laserprinters (fonts), kopieerapparaten en faxmachines.

6.15 Werking en principes van flash-geheugens



Figuur 3/6.15-5: Het wissen van een "5 V-only/negatieve gate flash EPROM".

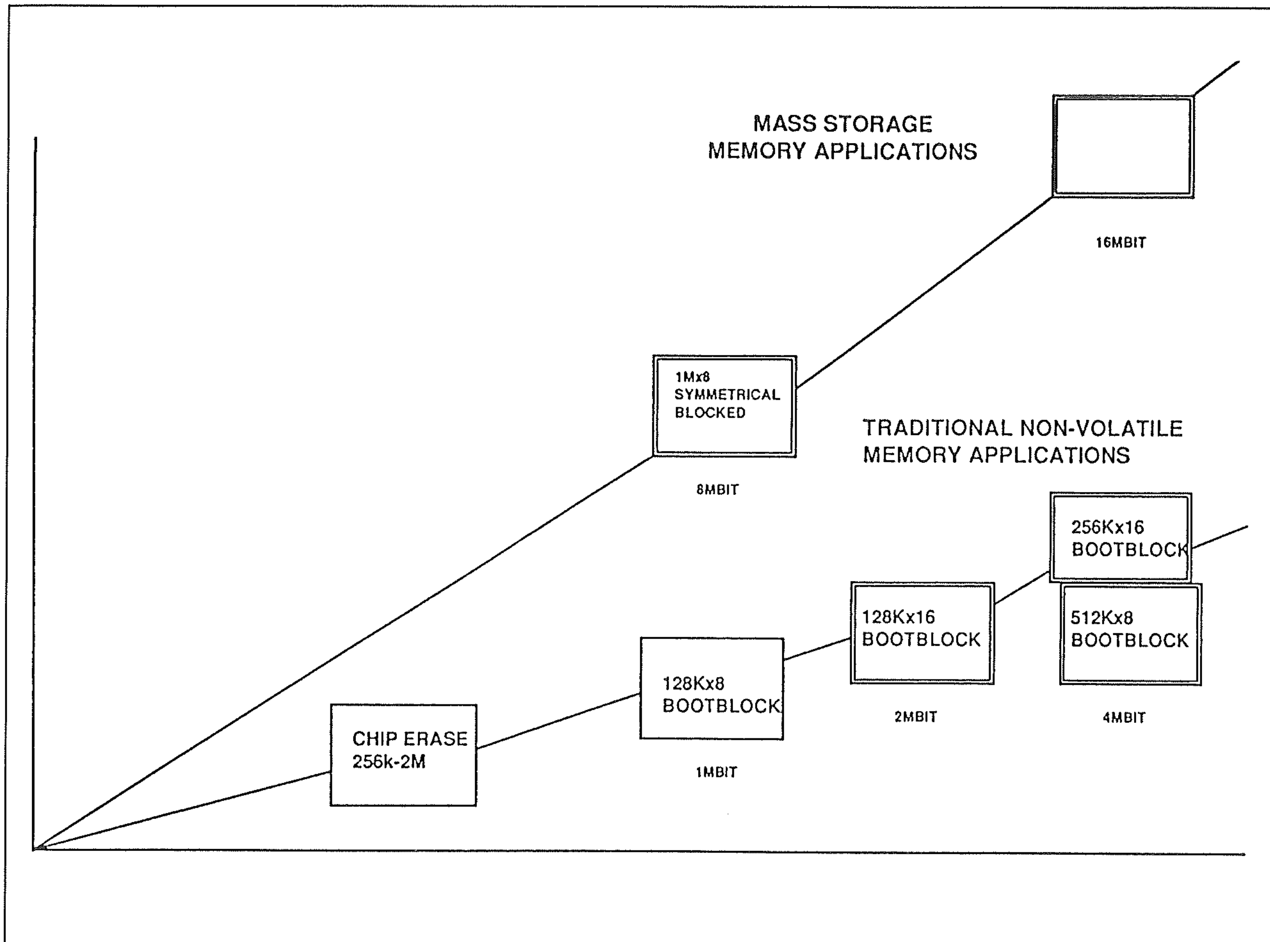
De flash-geheugens koppelen niet-vluchtigheid aan de mogelijkheid om meer dan 10.000 maal elektrisch te wissen en te programmeren. Hierdoor zijn deze IC's een aantrekkelijk alternatief voor schijf, EEPROM en batterijgevoede statische RAM. In gevallen waar periodieke aanpassing van code en datatabellen nodig is, is dit geheugen een ideale vervanging voor EPROM.
Wanneer primaire toepassingen en operating systemen in flash worden opgeslagen vervalt het langzame disk-DRAM download proces, waardoor het prestatievermogen van het systeem drastisch zal verbeteren, terwijl het opgenomen vermogen flink zal dalen.

Door het elektrisch wissen en in-system updaten neemt de flexibiliteit aanmerkelijk toe.
In schijfloze werkstations en terminals wordt het verkeer over de netwerken tot een minimum beperkt en kunnen de systemen zeer snel opstarten. Bij onderbrekingen van de netspanning worden tijdrovende "re-boot"-perioden dan ook vermeden.

Voor ingebedde systemen biedt een flash-IC betere prestaties, een lager energieverbruik, onmiddellijk inschakelen en een "execute in place" geheugen-hiërarchie voor het inlezen van code en data. Bovendien is het flash-geheugen betrouwbaarder en robuuster voor toepassingen in agressieve omgevingen. Speciaal voor "upgradeble" BIOS-toepassingen heeft Intel een flash-geheugen met "boot block" ontwikkeld.
Hiermee kan de BIOS veilig worden veranderd, zonder de computer open te maken, met behulp van een floppy-disk of via een modem per telefoon.

Flash hard disks
Een belangrijke toekomstige toepassing van flash-geheugens mag niet onvermeld blijven.
Om uitwisselbare "flash hard-disks" te verkrijgen werken fabrikanten samen om tot echte standaardisatie van de connectoren te komen. Op dit moment zijn al twee normen voor de 68-polige connector in gebruik: PCMIA (Personal Computer Memory Card International Association) en JEIDA (Japanese Electronic Industry Development Association). De toekomstige norm - ExCA - moet geschikt zijn voor uitwisselbare geheugen-, modem-, fax- en LAN-insteekkaarten en/of 1 inch hard disk-drives.

6.15 Werking en principes van flash-geheugens



Figuur 3/6.15-6: Verwachte ontwikkelingen op het gebied van de flash-geheugens.

Toekomstperspectieven
Het zal duidelijk zijn dat flash-geheugens een grote toekomst hebben!
Figuur 3/6.15-6 geeft een indruk van de door Intel verwachte ontwikkelingen op het gebied van de enkele-cel flash-geheugenchips.

Een voorbeeld: de 28F256

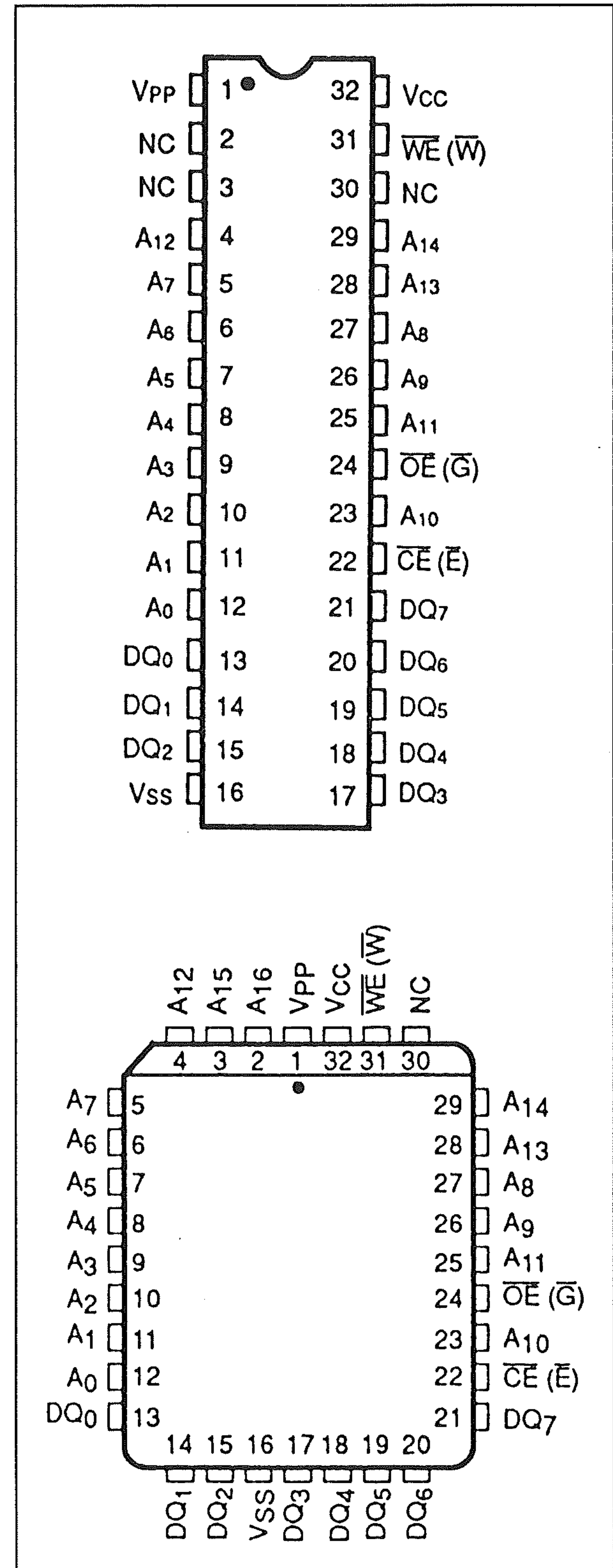
Kennismaking
De 28F256 is een 256 kB "flash"-type elektrisch wisbaar, elektrisch programmeerbaar read-only geheugen met een 32 kB x 8 bit organisatie.
Dit geheugen kan zowel in een EPROM-programmer als "in-circuit" worden ge(her)programmeerd en is verkrijgbaar in 32 pens plastic, keramische DIL of LCC behuizing.
De 28F256 heeft toegangstijden van minimaal 120 ns (AMD-type: 90 ns), waardoor toepassing met high-speed microprocessoren zonder wait-states mogelijk is.
Om bus-rivaliteit te voorkomen heeft dit niet-vluchtige geheugen aparte chip-enable ($\overline{CE}$) en output-enable ($\overline{OE}$) ingangen.
De chip combineert de functionaliteit van een EPROM met de mogelijkheid van elektrisch wissen en programmeren zonder de chip uit de schakeling te halen.

6.15 Werking en principes van flash-geheugens

Om dit alles mogelijk te maken in een 32-pens behuizing is de 28F256 voorzien van een Command Register.
De 28F256 kan minstens 10.000 maal betrouwbaar worden gewist en geprogrammeerd. Latch-up wordt voorkomen voor belastingen tot 100 mA op alle pennen, tussen -1 V en V_{CC} + 1 V.

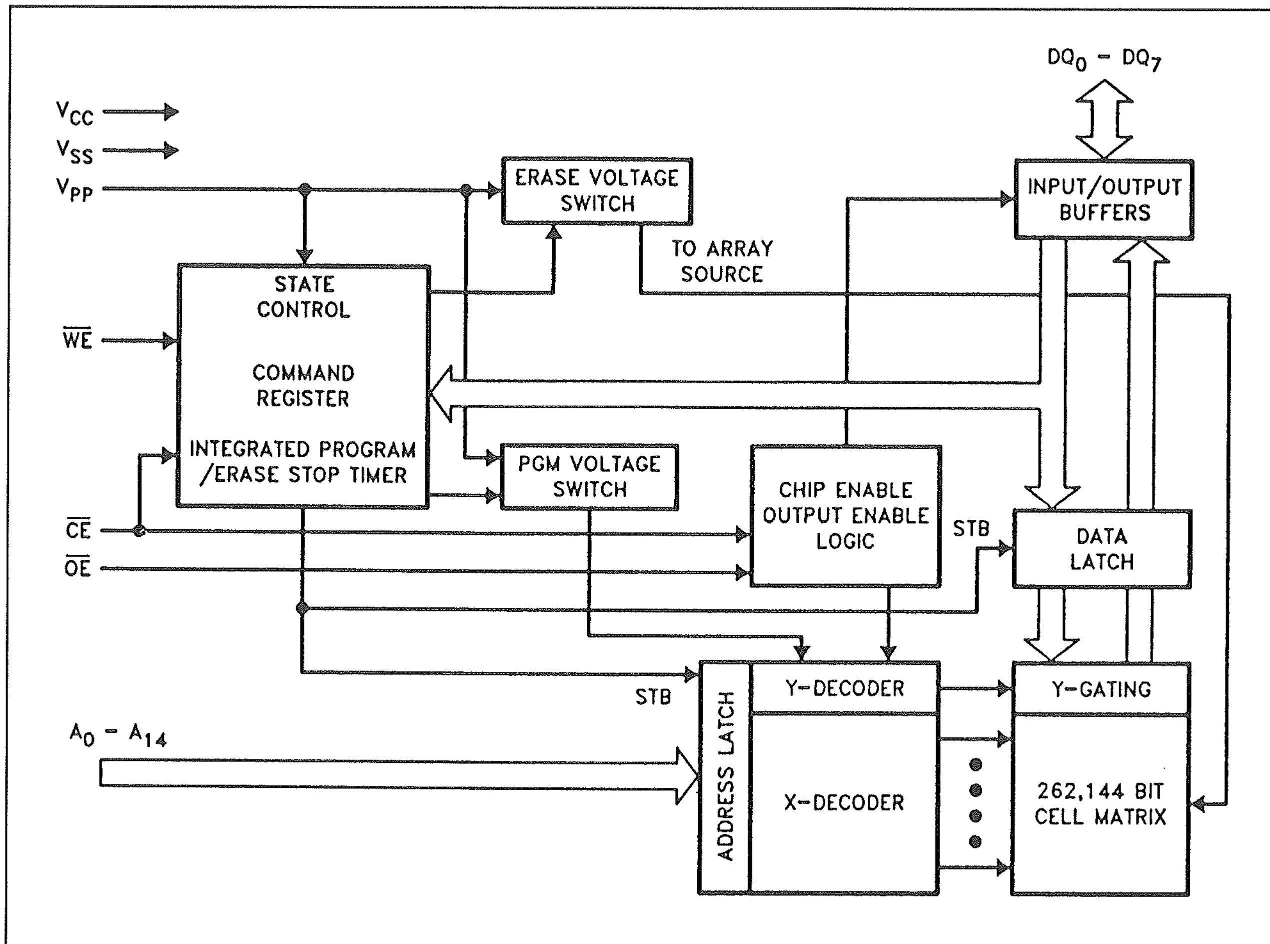
Specificaties
De 28F256 heeft de volgende algemene kenmerken:
- flash elektrische Chip-Erase typisch 1 s;
- Quick-Pulse programmeer-algorithme: byte programmeren in typisch 10 µs; chip programmeren in typisch 0,5 s;
- 10 000 wis/programmeercycli minimaal;
- programmeer/wisspanning V_{PP} van 12 V +/-5 %;
- toegangstijd 120 ns (AMD-type 90 ns minimaal);
- CMOS dissipatie: 10 mA aktief, 50 µA standby;
- geïntegreerde programmeer/wis stoptimer;
- Command Register architectuur voor microprocessor/microcontroller compatibele schrijf-interface;
- on-chip adres- en data-latches;
- één-transistor geheugencellen
- latch-up bescherming tot 100 mA van -1 V tot V_{CC} +1 V;
- behuizingen:
  JEDEC standaard plastic;
  keramische 32-pens;
  32-pens LCC;
- fabrikanten onder andere: Intel (28F256A), AMD en Toshiba (TC58257A).

De aansluitgegevens en het interne blokschema van de 28F256 zijn weergegeven in de figuren 3/6.15-7 en -8.

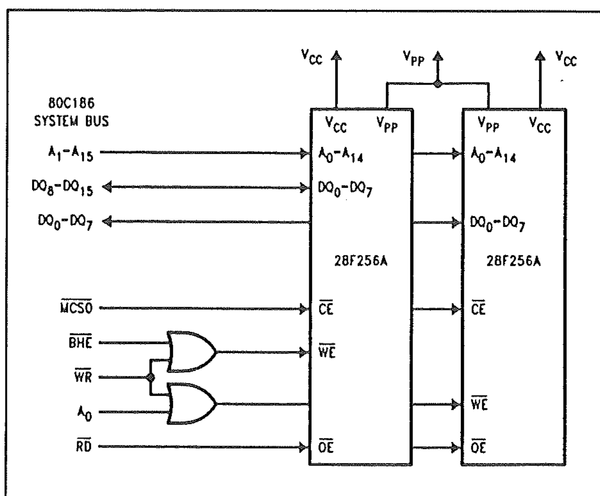


Figuur 3/6.15-7: De behuizingen en aansluitgegevens van de 28F256.

6.15 Werking en principes van flash-geheugens



Figuur 3/6.15-8: Het interne blokschema van de 28F256.



Figuur 3/6.15-9: Aansluiting van twee 28F256A's op een 80C186 systeem.

De 28F256 heeft voorzieningen die het aansluiten op een microprocessor gemakkelijk maken. In figuur 3/6.15-9 is geschetst hoe twee 28F256 geheugens op de systeembus van een 80C186 microcontroller kunnen worden aangesloten. Door de architectuur van de 28F256 zijn zeer weinig interface-schakelingen nodig voor complete in-circuit updates van de geheugeninhoud.

Beschrijving van de pennen

Een korte functie-beschrijving van de belangrijkste aansluitingen van dit IC.

– A0 tot en met A14

6.15 Werking en principes van flash-geheugens

De adres-ingangen, die intern worden gelatched gedurende schrijf-cycli.

- DQ0 tot en met DQ7
  De data in- en uitgangen. Deze pennen dienen als gegevensinvoer gedurende schrijf-cycli en gegevensuitvoer tijdens lees-cycli. De data-pennen zijn actief "HOOG" en gaan naar "TRI-STATE" als de chip wordt gede-activeerd of de uitgangen worden ge-disabled. De gegevens worden intern gelatched tijdens schijf-cycli.
- $\overline{CE}$
  De Chip Enable ingang activeert de interne controle-logica, de ingangsbuffers, de decoders en de sense-versterkers. Deze ingang is actief "LAAG", een "H" op deze ingang schakelt de chip uit en reduceert het stroomniveau tot "stand-by level".
- $\overline{OE}$
  De Output Enable is een actief "LAAG" ingang, die de poorten bestuurt tussen de data-uitgangen en de data-buffers.
- $\overline{WE}$
  Deze Write Enable ingang controleert het schrijven van gegevens naar het array door het besturen van het interne controle-register. Deze ingang is actief "LAAG". Adressen worden gelatched op de dalende flank en de gegevens op de stijgende flank van het $\overline{WE}$-signaal. Dit geldt echter alleen als de spanning op V_{PP} groter is dan 6,5 V. Is deze spanning lager, dan kan de inhoud van het geheugen niet worden gewijzigd.
- V_{PP}
  Dit is de voedingsspanning voor het wissen en programmeren van het geheugen-array. De spanning op deze pen moet groter zijn dan 6,5 V.
- V_{CC}
  De voedingsspanning van het IC, gelijk aan 5 V +/-10 %.
- V_{SS}
  De massa-aansluiting van de schakeling.

Werking van de 28F256

Het flash-geheugen 28F256 combineert de functies van een EPROM met de mogelijkheid om elektrisch te wissen en te programmeren. De 28F256 is hiertoe voorzien van een Command Register dat 100 % TTL-compatibele besturingssignalen, een gefixeerde voedingsspanning tijdens wissen en programmeren en maximale EPROM-compatibiliteit mogelijk maakt.
Wanneer de hoge spanning op de programmeerpen (12 V op V_{PP}) ontbreekt werkt de 28F256 als een ROM. Door signalen op de externe memory-control pennen worden dan de standaard EPROM lees, standby, output disable en intelligent identifier functies uitgevoerd.
Dezelfde EPROM lees, standby en output disable operaties zijn ook beschikbaar als de 12 V programmeerspanning wél op de V_{PP}-pen staat. Bovendien wordt in dat geval wissen en programmeren van het IC mogelijk. Het Command Register is dan vrijgegeven. Alle functies die betrekking hebben op het veranderen van de inhoud van het geheugen, met name intelligent identifier, wissen, wis/verifieer, programmeren en programmeer/verifieer, zijn toegankelijk via het Command Register.
De commando's worden met behulp van standaard microprocessor schrijftimingen naar het register geschreven. De inhoud van het register dient als instelling van een interne state-machine die de schakelingen voor het wissen en programmeren bestuurt. Met schrijfcycli worden ook de adressen en data die voor het wissen en programmeren nodig zijn intern gelatched. Wanneer het juiste commando in

het register is geschreven kan array-data, de intelligente identificatiecode of uitgangsdata door de microprocessor worden uitgelezen voor verificatie.

Geïntegreerde programmeer/wis Stop Timer

De tijdsduur van de programmeer- en wisoperaties wordt bepaald door achtereenvolgende commando-schrijfcycli. Bovendien worden de wis- en programmeercycli gewoonlijk gevolgd door de bijbehorende verificatie-commando's. De timing van deze handelingen wordt door een geïntegreerde stop timer (zie blokschema, figuur 3/6.15-8) vergemakkelijkt, waardoor programmeer/wis-specificaties overbodig worden. De tijdsduren voor wissen en programmeren zijn hierdoor minimaal.
Als de stop-timer een programmeer- of wis-operatie beëindigt, komt het geheugen in een niet-aktieve toestand totdat een verifieer- of reset-commando wordt ontvangen.

Schrijf-beveiliging

Het Command Register is alleen aktief als op V_{pp} een hoge spanning (12 V nominaal) aanwezig is.
Afhankelijk van de toepassing kan de ontwerper ervoor kiezen om de V_{pp}-voeding afschakelbaar te maken, waardoor die alleen beschikbaar is als de inhoud van het geheugen gewijzigd moet worden.
Als $V_{pp} = V_{pp\text{-}L}$ komt de inhoud van het register overeen met het lees-commando, waardoor de 28F256 als ROM werkt. De inhoud van het geheugen kan dan niet worden veranderd.
De ontwerper kan V_{pp} echter ook continu aangesloten laten. In dat geval worden de functies van het Command Register gesperd als V_{cc} beneden de schrijf-lockout spanning V_{LKO} komt (zie Power Up/Down beveiliging). De 28F256 is geschikt voor beide manieren.

Lezen

De 28F256 heeft twee besturingsfuncties die beide logisch aktief moeten zijn om data aan de uitgangen te verkrijgen. Chip-Enable ($\overline{CE}$) is de besturing van de voeding en moet voor de selectie van de component worden gebruikt. Met Output-Enable ($\overline{OE}$) wordt de uitgang bediend en is het mogelijk om, onafhankelijk van de keuze van het IC, data van de uitgangspennen te halen.
Als V_{pp} HOOG is ($V_{pp\text{-}H}$) kunnen de leesoperaties worden gebruikt voor het ophalen van array-data, de intelligente identificatiecodes en data voor programmeer/wis-verificatie. Als V_{pp} LAAG is ($V_{pp\text{-}L}$) kan alleen de array-data worden uitgelezen.

Output Disable

Wanneer Output-Enable op een logisch-hoge waarde (V_{IH}) staat, worden de uitgangen van het geheugen gesperd. De uitgangspennen bevinden zich dan in een hoog-impedante toestand.

Standby

Als Chip-Enable op een logisch-hoge waarde staat, worden door de standby-werking de meeste schakelingen in de 28F256 gesperd, waardoor het opgenomen vermogen beduidend lager wordt. De uitgangen worden, onafhankelijk van het Output-Enable signaal, in een hoog-impedante toestand gezet. Als de 28F256 tijdens het wissen, programmeren of verificatie van programmeren/wissen gedeselecteerd wordt blijft het geheugen aktieve stroom trekken totdat de operatie is beëindigd.

6.15 Werking en principes van flash-geheugens

Operation	Pins	V_{PP}(1)	A_0	A_9	$\overline{CE}$	$\overline{OE}$	$\overline{WE}$	DQ_0–DQ_7
READ-ONLY	Read	V_{PPL}	A_0	A_9	V_{IL}	V_{IL}	V_{IH}	Data Out
	Output Disable	V_{PPL}	X(7)	X	V_{IL}	V_{IH}	V_{IH}	Tri-State
	Standby	V_{PPL}	X	X	V_{IH}	X	X	Tri-State
	int$_e$ligent ID Manufacturer(2)	V_{PPL}	V_{IL}	V_{ID}(3)	V_{IL}	V_{IL}	V_{IH}	Data = 89H
	int$_e$ligent ID Device(2)	V_{PPL}	V_{IH}	V_{ID}(3)	V_{IL}	V_{IL}	V_{IH}	Data = B9H
READ/ WRITE	Read	V_{PPH}	A_0	A_9	V_{IL}	V_{IL}	V_{IH}	Data Out(4)
	Output Disable	V_{PPH}	X	X	V_{IL}	V_{IH}	V_{IH}	Tri-State
	Standby(5)	V_{PPH}	X	X	V_{IH}	X	X	Tri-State
	Write	V_{PPH}	A_0	A_9	V_{IL}	V_{IH}	V_{IL}	Data In (6)

Figuur 3/6.15-10: De bus-operaties van de 28F256.

Intelligente Identificatie

Met de intelligente identificatie-operatie (ook auto-select genoemd) komt de fabrikantencode (Intel: 89H, AMD: 01H) en de device-code (Intel: B9H, AMD: A1H) beschikbaar. De intelligente programmeerapparatuur past de wis- en programmeer-algoritmen automatisch hierop aan. Met Chip-Enable en Output-Enable op een logisch-laag niveau wordt deze operatie geaktiveerd door A9 op een hoge spanning V_{ID} te brengen (zie de tabel van figuur 3/6.15-10). De data die van de lokaties 0000H (hexadecimaal 0000) en 0001H worden gelezen komen overeen met respectievelijk de fabrikantencode en de device-code.

Beide codes kunnen ook worden uitgelezen via het Command Register als de 28F256 bijvoorbeeld in het doelsysteem wordt gewist en opnieuw geprogrammeerd.

Na het schrijven van 90H in het Command Register komt de fabrikantencode (89H) op adres 0000H beschikbaar, terwijl op adres 0001H de device-code (B9H) kan worden uitgelezen.

Schrijven

Wissen en programmeren van het geheugen worden uitgevoerd via het Command Register als een hoge spanning op de V_{PP}-pen wordt gezet. De inhoud van het register dient dan als ingang voor de inwendige state-machine. De uitgangen van de state-machine bepalen vervolgens de werking van de schakeling.

Het Command Register bezet zélf geen adresseerbare geheugenlokatie. Het register is een latch die wordt gebruikt om het commando en de voor de uitvoering van het commando benodigde adres- en data-informatie op te slaan.

Het Command Register wordt beschreven door Write-Enable ($\overline{WE}$) op een logisch-laag niveau (V_{IL}) te brengen, terwijl Chip-Enable "LAAG" is. Adressen worden gelatched op de dalende flank van Write-Enable, terwijl data op de stijgende flank van de Write-Enable puls wordt gelatched.

Command	Bus Cycles Req'd	First Bus Cycle			Second Bus Cycle		
		Operation(1)	Address(2)	Data(3)	Operation(1)	Address(2)	Data(3)
Read Memory	1	Write	X	00H			
Read inteligent ID Codes	3	Write	X	90H	Read	(4)	(4)
Set-Up Erase/Erase(6)	2	Write	X	20H	Write	X	20H
Erase Verify(6)	2	Write	EA	A0H	Read	X	EVD
Set-Up Program/Program(5)	2	Write	X	40H	Write	PA	PD
Program Verify(5)	2	Write	X	C0H	Read	X	PVD
Reset(7)	2	Write	X	FFH	Write	X	FFH

Figuur 3/6.15-11: Definities van de commando's.

Hierbij zijn de standaard microprocessor timingen van kracht.

Definities van de Commando's

Wanneer op de V_{PP}-pen een lage spanning staat, wordt de inhoud van het Command Register automatisch (default) 00H, waardoor read-only operaties mogelijk worden.

Door 12 V op de V_{PP}-pen te zetten worden lees/schrijf-operaties toegestaan. Welke operaties dat zijn wordt bepaald door de data-patronen die in het Command Register worden geschreven. In de tabel van figuur 3/6.15-11 wordt een overzicht van de register-commando's gegeven, terwijl ze hieronder apart worden behandeld.

Lees-commando (Read Memory)

Terwijl V_{PP} "HOOG" is om te kunnen wissen en programmeren, kunnen de inhouden van het geheugen worden bereikt met het lees-commando. De lees-operatie wordt ingeleid door 00H in het Command Register te schrijven. Door microprocessor leescycli wordt array-data opgehaald. Het geheugen blijft bereikbaar voor uitlezen totdat de inhoud van het Command Register wordt veranderd.

Bij het opkomen van de voedingsspanning is de inhoud van het register automatisch (default) 00H. Hierdoor wordt voorkomen dat bij verschijnen van V_{PP} de inhoud van het geheugen per ongeluk verandert. Als V_{PP} altijd op de 28F256 aanwezig is ("hard-wired"), komt het IC op spanning en blijft beschikbaar voor uitlezen totdat de inhoud van het Command Register wordt veranderd.

Inlezen van de identificatie

Flash-geheugens zijn bedoeld voor toepassingen, waarbij de inhoud van het geheugen door de lokale CPU wordt veranderd. Daarom moet het mogelijk zijn de fabrikant- en device-codes uit te lezen terwijl de component zich in het definitieve systeem bevindt. PROM-programmers verkrijgen de identificatiecodes meestal door A9 op een hoge spanning te brengen. In de praktijk is het echter niet gewenst een hoge spanning naar adreslijnen te multiplexen.

De 28F256 kan een intelligente identificatie-operatie uitvoeren ("Read Intelligent ID Codes") die aan de traditionele PROM-programmeer methodologie wordt toegevoegd. Deze handeling wordt ingeleid

6.15 Werking en principes van flash-geheugens

door 90H in het Command Register te schrijven. Na deze schrijf-operatie levert een leescyclus op adres 0000H de fabrikant-code 89H (Intel) op. Op adres 0001H kan vervolgens de device-code (B9H) worden uitgelezen. Om deze operatie te beëindigen is het nodig om een ander geldig commando in het register te schrijven.

Set-up wis/wis
(Set-up Erase/Erase) commando's

Set-up erase is een commando waardoor het geheugen wordt klaargezet voor wissen van alle bytes in het array. Deze operatie wordt uitgevoerd door 20H in het Command Register te schrijven. Om het wissen van de chip te beginnen moet het wis-commando (20H) nog een keer in het register worden geschreven.
Het wissen begint dan op de stijgende flank van de Write-Enable puls en eindigt op de stijgende flank van de volgende Write-Enable puls van bijvoorbeeld het Wis-Verifieer commando. Door deze tweetraps volgorde van set-up, gevolgd door de uitvoering, wordt voorkomen dat de inhoud van het geheugen per ongeluk wordt uitgewist. Bovendien kan wissen van de chip alleen gebeuren als op de V_{PP}-pen een hoge spanning staat. Bij afwezigheid van deze spanning is de inhoud van het geheugen beveiligd tegen wissen.

Wis-Verifieer (Erase-Verify) commando

Met het wis-commando worden alle bytes van het array parallel gewist. Na elke wis-operatie moeten alle bytes worden geverifieerd. De wis-verifieer operatie wordt ingeleid door A0H in het Command Register te schrijven. Het adres van de byte die moet worden geverifieerd moet op de dalende flank van de Write-Enable puls worden gelatched. De wis-operatie wordt op de stijgende flank van de Write-Enable puls van het schrijven naar het register beëindigd. De 28F256 zet een inwendig opgewekte marge-spanning op de geadresseerde byte. Wanneer de inhoud van de geadresseerde byte FFH is, zijn alle bits hierin gewist. Het wis-verifieer commando moet in het Command Register worden geschreven voordat het adres voor de byte-verificatie wordt gelatched. Dit proces gaat door voor alle bytes in het array totdat een byte geen FFH oplevert of als het laatste adres is uitgelezen. In het geval dat de uitgelezen data niet FFH is wordt nog een wis-operatie uitgevoerd (zie ook Wis set-up/wis). Er wordt dan geverifieerd vanaf de laatste geverifieerde byte. Zijn alle bytes in het array geverifieerd dan is de wis-operatie klaar en kan het IC worden geprogrammeerd. Op dit punt wordt het verifiëren beëindigd door een geldig commando (bijvoorbeeld Program Set-up) in het Command Register te schrijven. In figuur 3/6.15-12 (het "Quick Puls Algorithme") is te zien hoe commando's en bus-operaties worden gecombineerd om de 28F256 elektrisch te kunnen wissen.

Set-up programmeer/
programmeer commando's

Set-up Program is een commando waardoor de component wordt klaargezet voor programmeren van de bytes. De set-up operatie wordt uitgevoerd door 40H in het Command Register te schrijven.
Nadat de set-up is uitgevoerd maakt de volgende Write-Enable puls dat aktief kan worden geprogrammeerd. De adressen worden inwendig op de dalende flank van de Write-Enable puls gelatched, terwijl data intern op de stijgende flank van de Write-Enable puls wordt gelatched. Op de stijgende flank van Write-Enable begint

ook het programmeren. De programmeer-operatie eindigt op de volgende stijgende flank van Write-Enable die wordt gebruikt om het program-verifieer commando te schrijven.

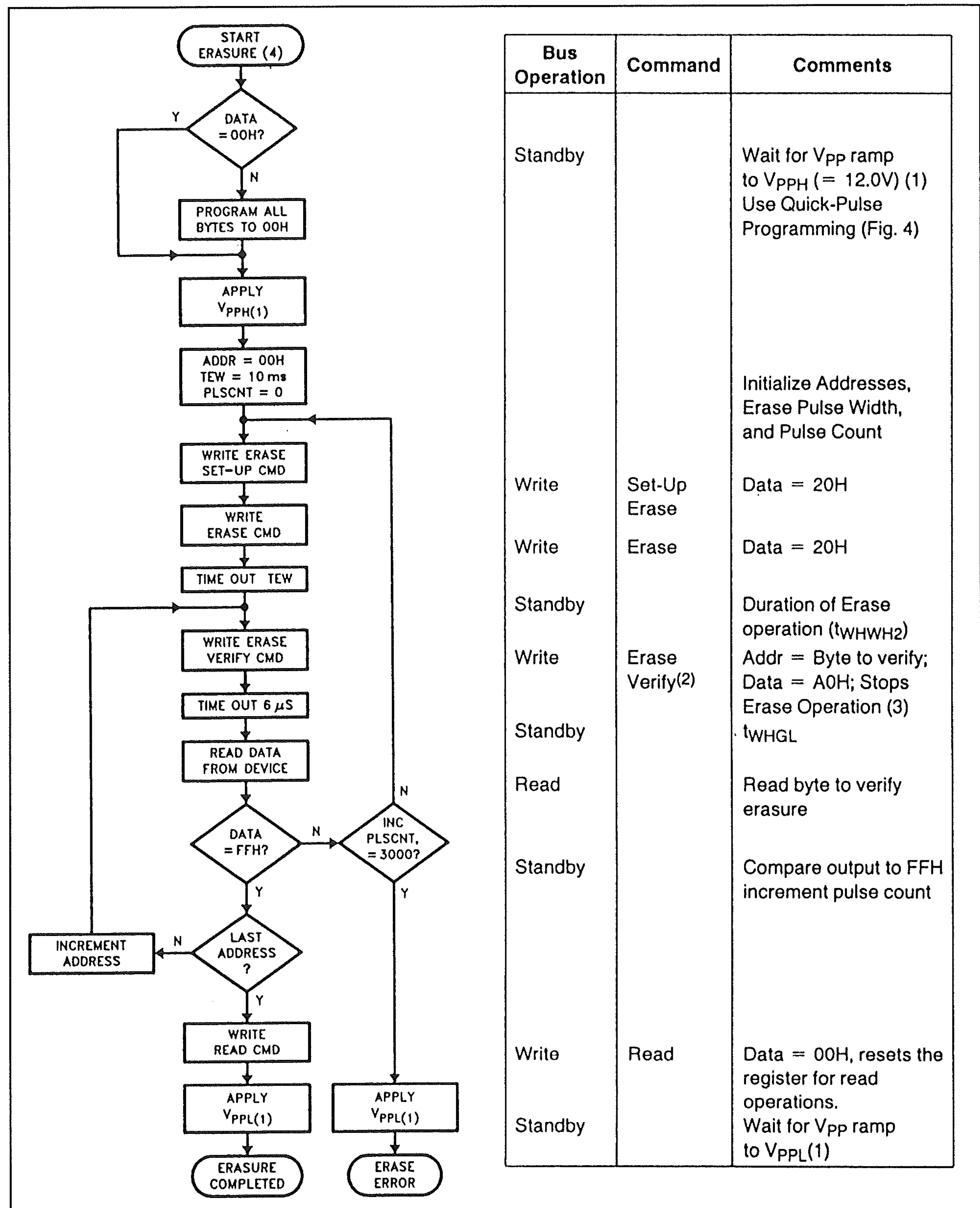
Program Verifieer commando
De 28F256 wordt geprogrammeerd op basis van byte-na-byte. Het byte-programmeren mag opeenvolgend of willekeurig gebeuren. Na iedere programmeer-operatie moet de net geprogrammeerde byte worden geverifieerd. De program-verifieer operatie wordt ingeleid door C0H in het Command Register te schrijven. Door het schrijven in het register wordt de programmeer-operatie op de stijgende flank van de Write-Enable puls beëindigd. De program-verifieer operatie zet de component klaar voor verificatie van de laatst geprogrammeerde byte. Er wordt geen nieuwe adres-informatie gelatched.
De 28F256 zet een inwendig opgewekte marge-spanning op de byte. De data verschijnt op een microprocessor lees-cyclus. Wanneer de geprogrammeerde data overeenkomt met de aangeboden data is de byte goed geprogrammeerd. Het programmeren gaat dan verder op de volgende gewenste byte-lokatie.
In figuur 3/6.15-13 (het "Quick Puls Programmeer Algorithme") is geschetst hoe de commando's worden gecombineerd met bus-operaties om de byteprogrammering uit te voeren.

Reset commando
De wis- en programmeer handelingen kunnen veilig worden beëindigd met een reset commando. Door de set-up commando's van zowel wissen als programmeren te laten volgen door het schrijven van twee opvolgende FFH's wordt de operatie veilig afgebroken. De inhoud van het geheugen zal hierdoor niet worden veranderd. Hierna moet een geldig commando worden ingeschreven om de component in de gewenste toestand te plaatsen.

Langdurig wissen/programmeren
Gebruikers zijn altijd bezorgd geweest over het vaak wissen/programmeren van EEPROM's. Het sterke elektrische veld dat voor het tunnelen van dunne oxyde EEPROM's nodig is, kan het oxyde op zwakke plaatsen letterlijk verscheuren. Om dit te bestrijden hebben sommige fabrikanten redundantieschema's opgenomen, waardoor deze fouten tot onbelangrijke niveaus beperkt bleven. Voor redundantie is echter verdubbeling van de cel-afmetingen nodig, een dure oplossing! Door de toegepaste ETOX II flashgeheugen technologie van Intel is zeer vaak wissen/programmeren mogelijk zonder toename van de afmetingen van de geheugencellen. De 28F256A is dan ook gespecificeerd voor 10.000 programmeer/wis-cycli. Het geheugen wordt geprogrammeerd en gewist met behulp van Intel's Quick-Pulse Programming en Quick-Erase algoritmen.

Het "Quick-pulse Programmeer algorithme"
Bij de Quick-Pulse Programmeer algorithme (zie figuur 3/6.15-13) zijn de programmeer-operaties 10 µs lang. Iedere operatie wordt gevolgd door een byteverificatie om te bepalen of de geadresseerde byte met succes is geprogrammeerd. De algorithme staat maximaal 25 programmeer-operaties per byte toe, hoewel de meeste bytes op de eerste of tweede operatie al goed zijn. De gehele volgorde van programmeren en byte verifiëren wordt uitgevoerd terwijl V_{PP} op een 12 V spanning staat.

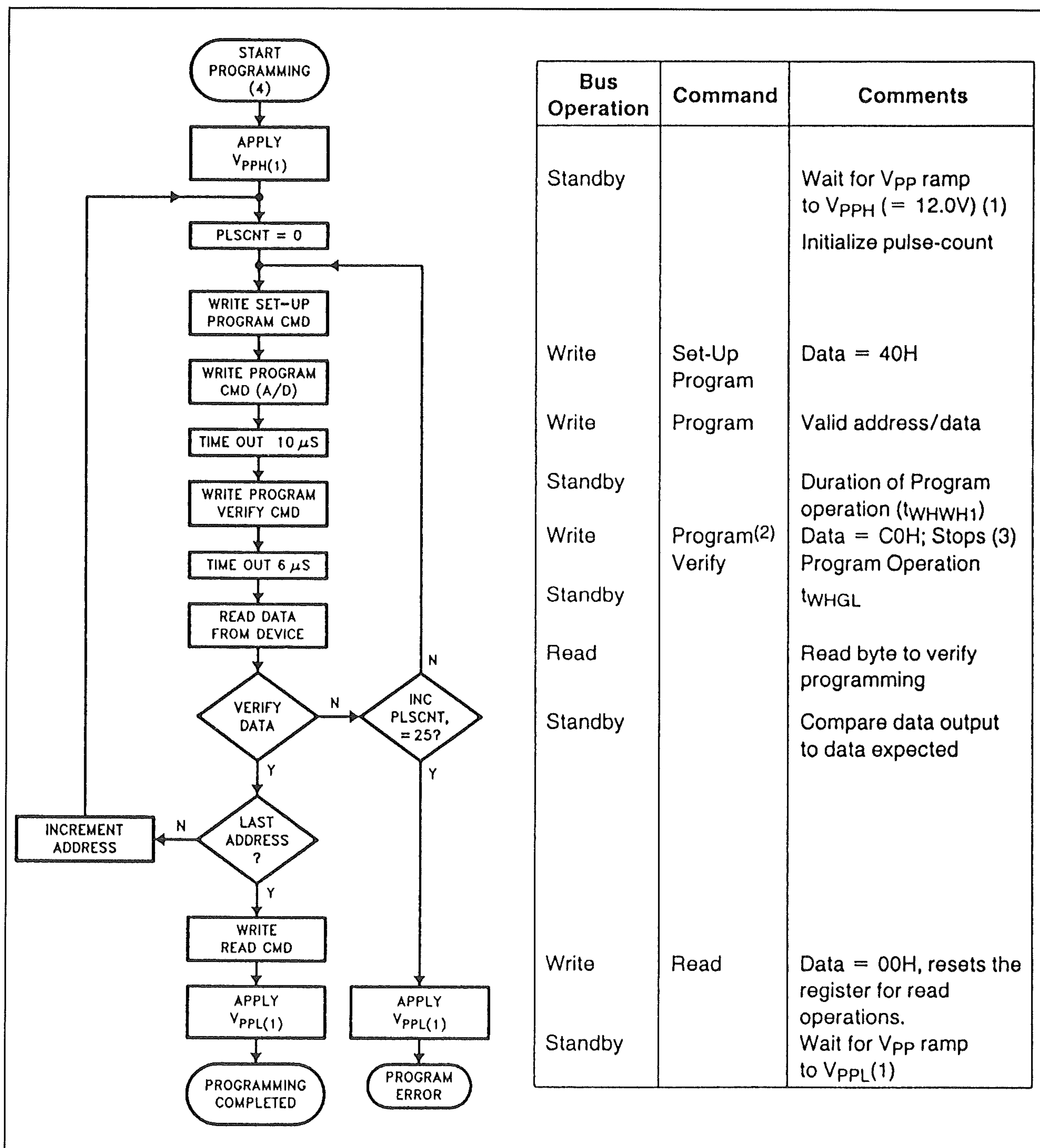
6.15 Werking en principes van flash-geheugens



Bus Operation	Command	Comments
Standby		Wait for V_{PP} ramp to V_{PPH} (= 12.0V) (1) Use Quick-Pulse Programming (Fig. 4)
		Initialize Addresses, Erase Pulse Width, and Pulse Count
Write	Set-Up Erase	Data = 20H
Write	Erase	Data = 20H
Standby		Duration of Erase operation (t_{WHWH2})
Write	Erase Verify(2)	Addr = Byte to verify; Data = A0H; Stops Erase Operation (3)
Standby		t_{WHGL}
Read		Read byte to verify erasure
Standby		Compare output to FFH increment pulse count
Write	Read	Data = 00H, resets the register for read operations.
Standby		Wait for V_{PP} ramp to V_{PPL}(1)

Figuur 3/6.15-12: Dit "Quick Pulse Algorithme" kan op de 28F256 worden toegepast.

6.15 Werking en principes van flash-geheugens



Bus Operation	Command	Comments
Standby		Wait for V_{PP} ramp to V_{PPH} (= 12.0V) (1) Initialize pulse-count
Write	Set-Up Program	Data = 40H
Write	Program	Valid address/data
Standby		Duration of Program operation (t_{WHWH1})
Write	Program(2) Verify	Data = C0H; Stops (3) Program Operation
Standby		t_{WHGL}
Read		Read byte to verify programming
Standby		Compare data output to data expected
Write	Read	Data = 00H, resets the register for read operations.
Standby		Wait for V_{PP} ramp to V_{PPL}(1)

Figuur 3/6.15-13: Het "Quick-Pulse Programmeer Algorithme" voor de 28F256.

Het "Quick-Erase algorithme"

Met het "Quick-Erase algorithme" wordt de inhoud van het geheugen snel en betrouwbaar elektrisch gewist. De algorithme verloopt (net als het "Quick-Pulse Programmeer algorithme") volgens een gesloten lus om simultaan de lading van alle bits in het array te verwijderen. Het wissen begint met het lezen van de geheugen-inhoud. De 28F256 wordt leeg gele-

6.15 Werking en principes van flash-geheugens

verd. Uitlezing van FFH-data kan direkt worden gevolgd door programmeren van het IC.
Schakelingen die gewist en geprogrammeerd zijn kunnen uniform en betrouwbaar worden gewist door eerst alle bits naar de geladen toestand (00H) te programmeren. Dit wordt met gebruik van het "Quick-Pulse Programmeer algorithme" in ongeveer een halve seconde uitgevoerd. De wis-operatie gaat dan door met een initiële wis-operatie. Verificatie van het wissen (data = FFH) begint op adres 0000H en gaat door tot het laatste adres of totdat van FFH afwijkende data wordt ontmoet. Het wissen kan efficiënter worden door het adres van de laatste geverifieerde byte in een register op te slaan. Na de volgende wis-operatie begint het verifiëren dan op het opgeslagen adres. Het wissen geschiedt in ongeveer één seconde.

Maatregelen bij het ontwerpen
Bij het ontwerpen van schakelingen waarin dit flash-EPROM wordt toegepast moet men rekening houden met de volgende speciale maatregelen.

- Tweelijns uitgangsbesturing
  Flash-geheugens worden vaak toegepast in grotere geheugen-array's. Om hieraan tegemoet te komen is de 28F256 uitgerust met twee read-control ingangen. Om deze besturingslijnen efficiënt te gebruiken moet een adresdecoder de Chip-Enable besturen, terwijl het lees-signaal van het systeem alle flash-geheugens en andere parallel geschakelde geheugens bestuurt.
  Hierdoor wordt gegarandeerd dat alleen data van vrijgegeven geheugens beschikbaar komt en dat de niet-geselecteerde geheugens in de standby toestand blijven.
- Ontkoppeling van de voeding
  Het aan- en uitschakelen van de voeding van flash-geheugens maakt zorgvuldige ontkoppeling noodzakelijk. Systeem-ontwerpers krijgen te maken met drie I_{CC}-gevallen: standby, aktief en pieken op de flanken van Chip-Enable. De hoogten van deze pieken zijn afhankelijk van de capacitieve en inductieve belastingen van de uitgangen.
  Spanningspieken worden onderdrukt door de tweelijns besturing en een juiste keuze van de ontkoppel-condensator. Elke component moet een keramische condensator van 0,1 µF tussen V_{CC} en V_{SS} en tussen V_{PP} en V_{SS} hebben. Plaats deze condensatoren zo dicht mogelijk bij de component! Bovendien moet per acht componenten een 4,7 µF elektrolytische condensator tussen V_{CC} en V_{SS} worden opgenomen.
- V_{PP}-lijn op printkaarten
  Voor het programmeren van flash-geheugens die zich in de definitieve schakeling bevinden is het nodig dat de ontwerper aandacht besteed aan het spoor voor V_{PP} in de gedrukte bedrading. Het wordt aanbevolen dezelfde spoorbreedten en layout-overwegingen te gebruiken als voor de V_{CC}-bus om spanningspieken en overshoots te vermijden.

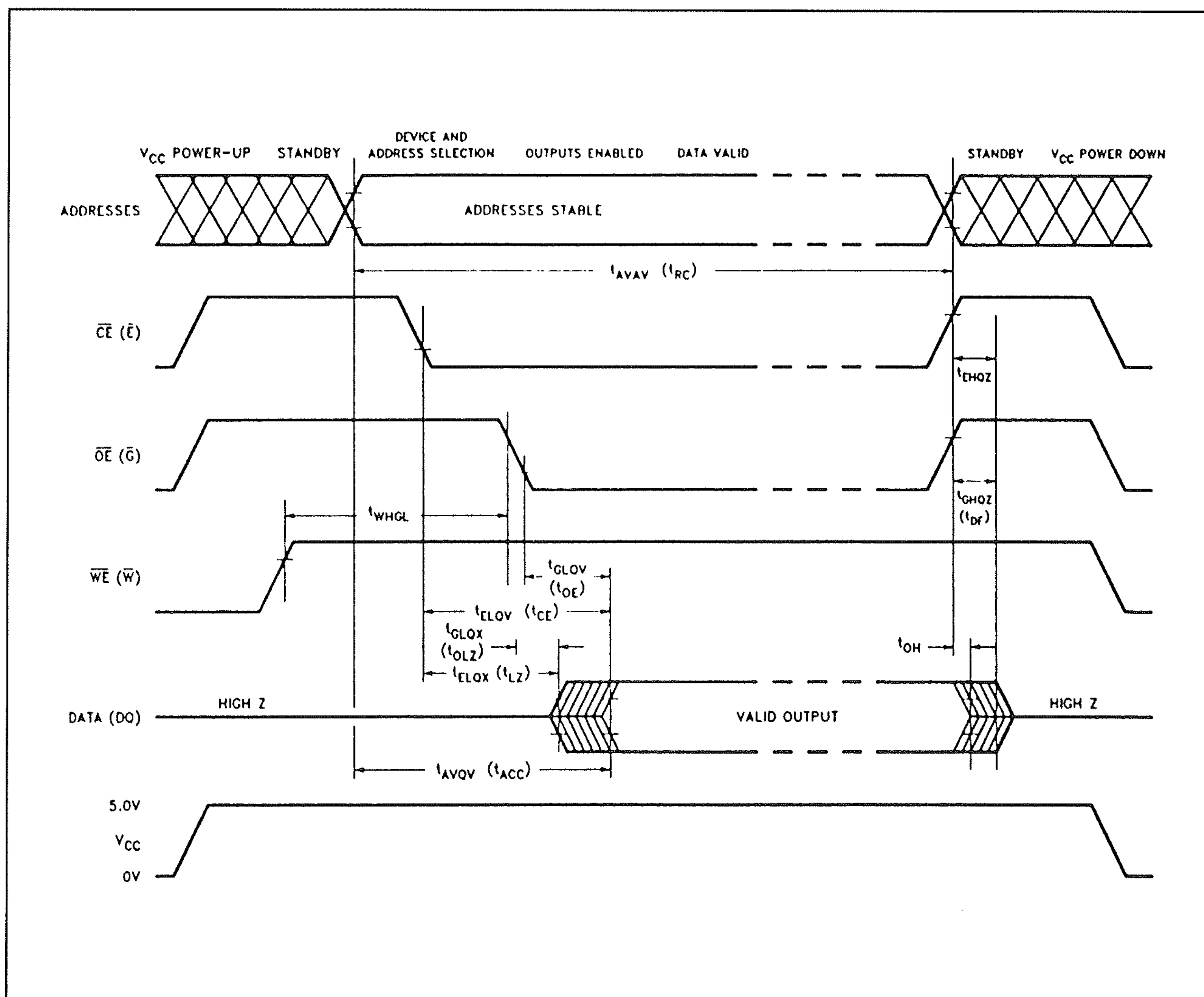
Power Up/Down beveiliging
Het ontwerp van de 28F256 biedt bescherming tegen per ongeluk wissen of programmeren bij veranderingen van de voedingsspanning.
Bij het inschakelen van de voeding maakt het voor de 28F256 niet uit of V_{CC} of V_{PP} het eerst aanwezig is. Inwendige schakelingen zorgen ervoor dat het Command Register bij power-up in de leesmode wordt gezet.

6.15 Werking en principes van flash-geheugens

De systeem-ontwerper moet ervoor waken dat wordt geschreven met V_{CC}-spanningen die hoger zijn dan V_{LKO} als V_{PP} aktief is. Aangezien zowel $\overline{WE}$ als $\overline{CE}$ "LAAG" moeten zijn voor een schrijfcommando, wordt schrijven voorkomen door één van beide "HOOG" te maken. De architectuur van het control register zorgt voor nog een extra beveiliging aangezien verandering van de inhoud van het geheugen alleen plaats vindt na succesvolle afsluiting van de tweetraps commando-volgorden.

Operation	Power Dissipation (Watt-Seconds)
Array Program/Program Verify	0.043
Array Erase/Erase Verify	0.083
One Complete Cycle	0.169

Figuur 3/6.15-14: Opgenomen vermogen bij het updaten van de 28F256A.



Figuur 3/6.15-15: Golfvormen bij uitlezen van de 28F256.

6.15 Werking en principes van flash-geheugens

Versions		Notes	28F256A-120		28F256A-150		28F256A-200		Unit
Symbol	Characteristic		Min	Max	Min	Max	Min	Max	
t_{AVAV}/t_{WC}	Write Cycle Time		120		150		200		ns
t_{AVWL}/t_{AS}	Address Set-Up Time		0		0		0		ns
t_{WLAX}/t_{AH}	Address Hold Time		60		60		75		ns
t_{DVWH}/t_{DS}	Data Set-Up Time		50		50		50		ns
t_{WHDX}/t_{DH}	Data Hold Time		10		10		10		ns
t_{WHGL}	Write Recovery Time before Read		6		6		6		μs
t_{GHWL}	Read Recovery Time before Write		0		0		0		μs
t_{ELWL}/t_{CS}	Chip Enable Set-Up Time before Write		20		20		20		ns
t_{WHEH}/t_{CH}	Chip Enable Hold Time		0		0		0		ns
t_{WLWH}/t_{WP}	Write Pulse Width	2	60		60		60		ns
t_{WHWL}/t_{WPH}	Write Pulse Width High		20		20		20		ns
t_{WHWH1}	Duration of Programming Operation	3	10		10		10		μs
t_{WHWH2}	Duration of Erase Operation	3	9.5		9.5		9.5		ms
t_{VPEL}	V_{PP} Set-Up Time to Chip Enable Low		1.0		1.0		1.0		μs

Figuur 3/6.15-16: Schakeltijden voor schrijven, wissen en programmeren van de 28F256.

Vermogensdissipatie van de 28F256A

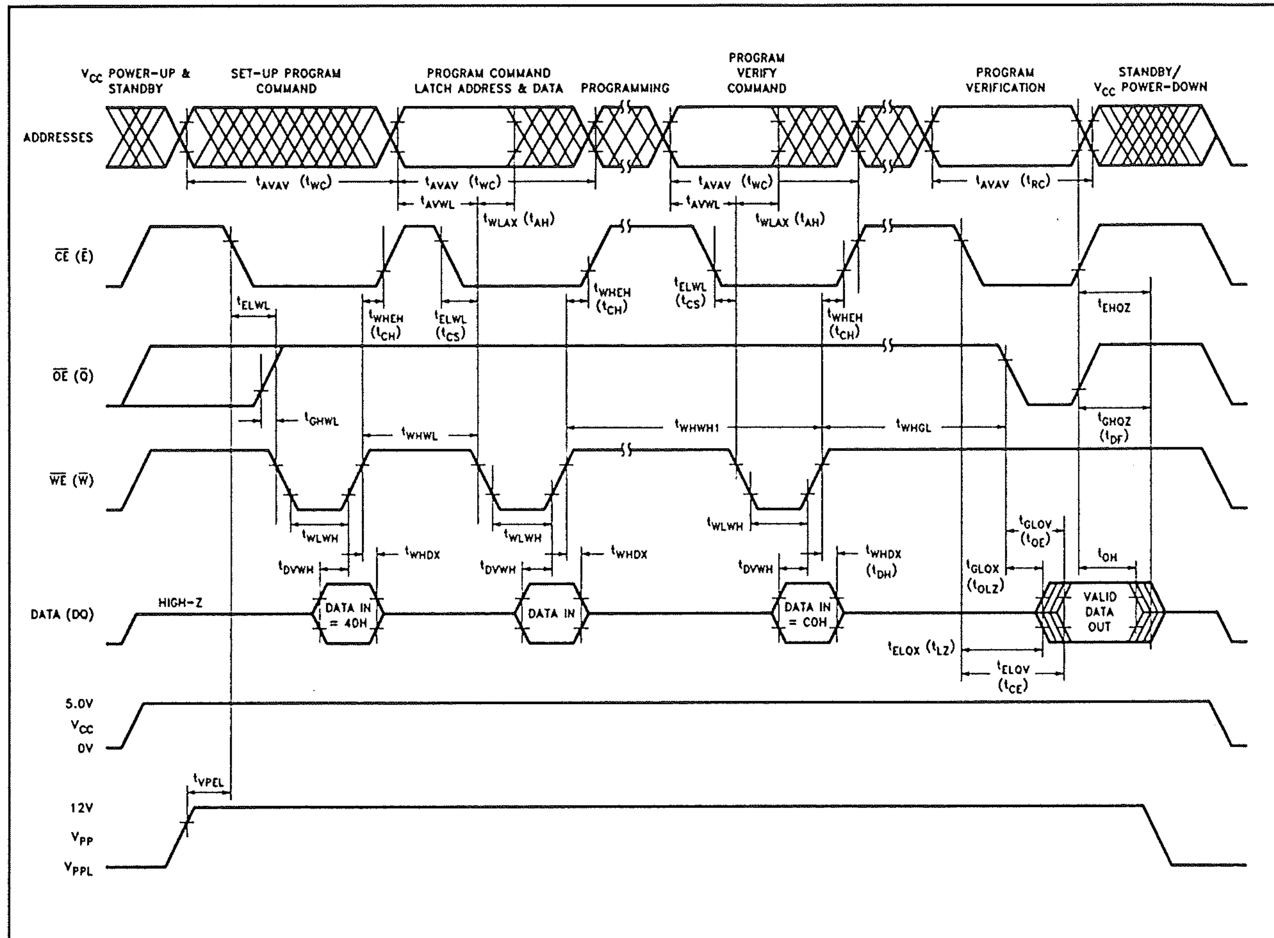
Wanneer draagbare systemen worden ontworpen moet de ontwerper rekening houden met het energieverbruik en dit niet alleen bij bedrijf maar ook bij afschakeling. De niet-vluchtige flash-geheugens verbruiken niets om de code of data vast te houden.

In figuur 3/6.15-14 is te zien hoeveel vermogen wordt gedissipeerd bij het updaten van de 28F256A.

De timing-karakteristieken

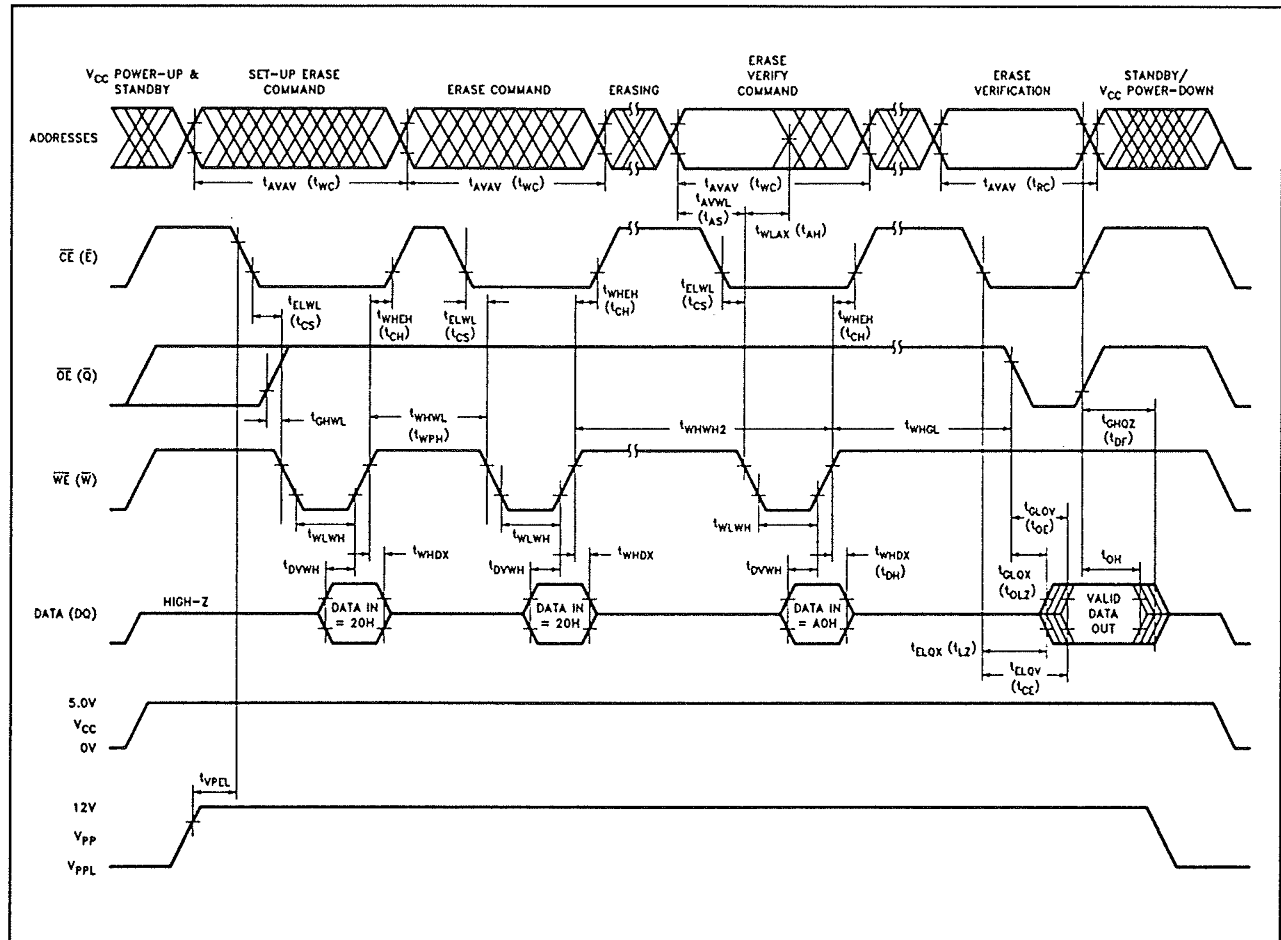
De timing-karakteristieken van het Inteltype van de 28F256A zijn opgenomen in de figuren 3/6.15-15 tot en met 3/6.15-18.

6.15 Werking en principes van flash-geheugens



Figuur 3/6.15-17: Golfvormen bij het programmeren van de 28F256.

6.15 Werking en principes van flash-geheugens



Figuur 3/6.15-18: Golfvormen bij het wissen van de 28F256.

6.15 Werking en principes van flash-geheugens

3/6.16

Digitale perifere drivers

Inleiding

Te klein vermogen

Met logische schakelingen kunnen allerlei regelingen en besturingen worden ontworpen. Vaak zijn echter de uitgangsstromen van de gebruikte logische families te klein en/of kunnen ze de benodigde uitgangsspanning niet aan. Bij TTL is bijvoorbeeld meestal maar een stroom van 16 mA naar aarde beschikbaar ("sink"-stroom), terwijl de uitgangsspanning ruwweg een zwaai kan maken tussen 0 en +5 V. Wanneer bijvoorbeeld een schakelklok voor fotografie (doka-timer) wordt gemaakt, kan de belichtingstijd nauwkeurig digitaal worden ingesteld, maar zal voor de op 220 V werkende lamp een apart relais nodig zijn. Ook wanneer een triac wordt gebruikt moet de aansturing hiervan eerst "vertaald" worden. Zelfs voor controlelampjes en LED's kan 5 V bij 16 mA onvoldoende zijn en is dan een extra drivertrap noodzakelijk. Ook wanneer men vanuit een microprocessor of een personal computer in "de buitenwereld" komt zijn vaak interfaceschakelingen nodig om voor de toepassing het benodigde vermogen te leveren. Bovendien is het voor de veiligheid van de computer beter om altijd van buffers gebruik te maken. Gaat er iets mis, dan "doet" de computer het in elk geval nog.

Digitale perifere drivers

Vroeger moesten deze tussentrappen opgebouwd worden met discrete schakelingen, dus met losse transistoren, weerstanden en dioden. Maar tegenwoordig bestaan er tientallen zogenoemde "digitale perifere drivers", waarin deze onderdelen geïntegreerd zijn.
Het ontwerpen van schakelingen wordt daardoor aanmerkelijk vereenvoudigd, terwijl vaak ook de print kleiner en minder complex worden.

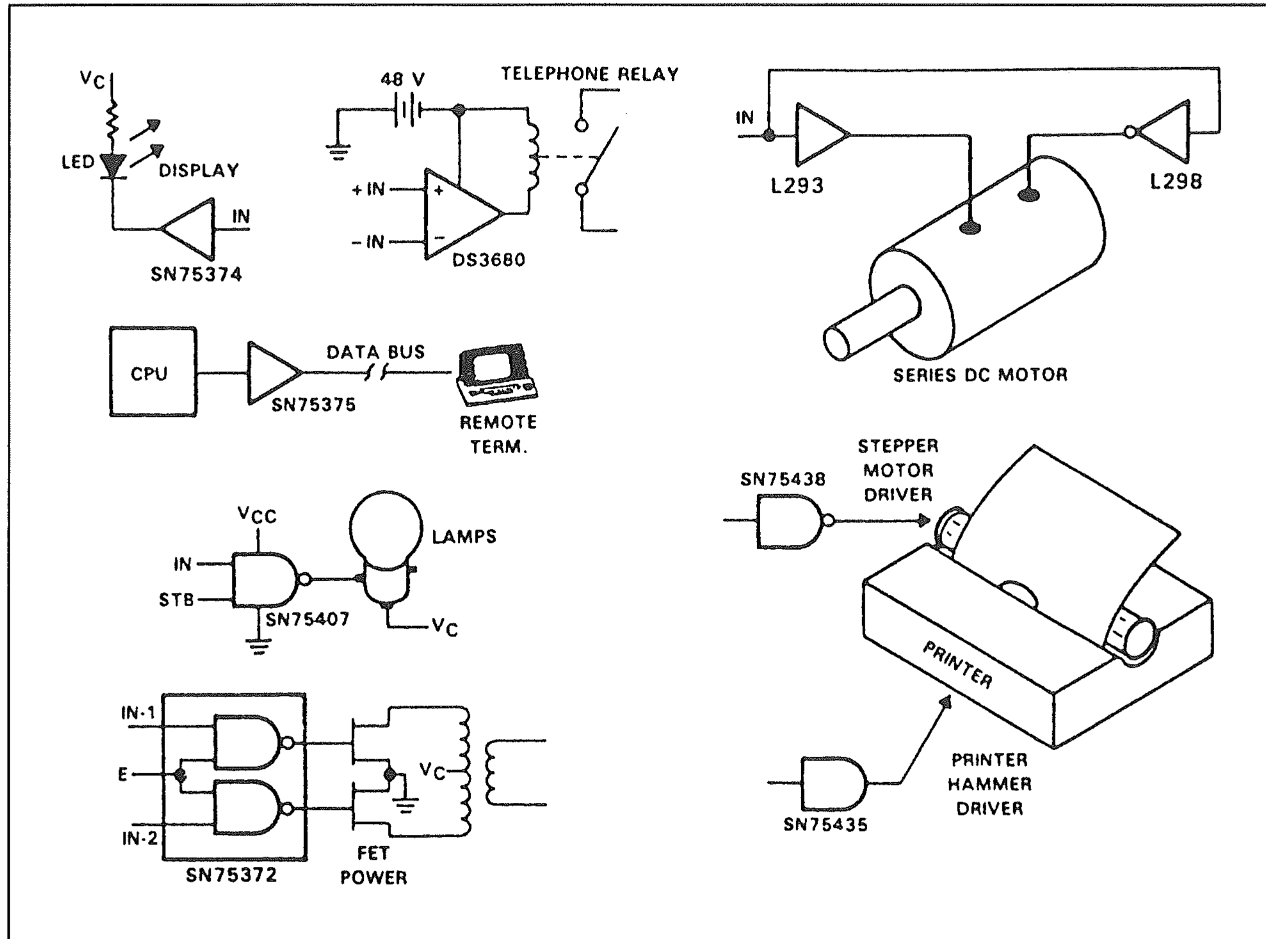
Digitale perifere drivers zijn geïntegreerde schakelingen die worden toegepast om de koppeling tussen signalen op TTL-, MOS- en CMOS-niveau en elektrische componenten met een grotere stroom en/of een hogere spanning direkt mogelijk te maken.
Het gaat dan bijvoorbeeld om lampjes, relais, spoelen, transmissielijnen en motoren.

In figuur 3/6.16-1 zijn een aantal toepassingen van digitale perifere drivers samengevat.

Samenstelling

De digitale perifere drivers zijn meestal voorzien van grotere uitgangstransistoren om het benodigde vermogen te leveren, voorafgegaan door een schakeling die het logische niveau moet verschuiven.

6.16 Digitale perifere drivers



Figuur 3/6.16-1: Enkele gevallen waarin digitale perifere drivers gebruikt moeten worden.

Als voorbeeld wordt in figuur 3/6.16-2a een digitale perifere driver getekend, waarbij weerstanden en een diode worden gebruikt om het niveau van het ingangssignaal geschikt te maken voor de uitgangsschakeling. In figuur 3/6.16-2b wordt dat met een logische poort gedaan.

Specificaties

Inleiding

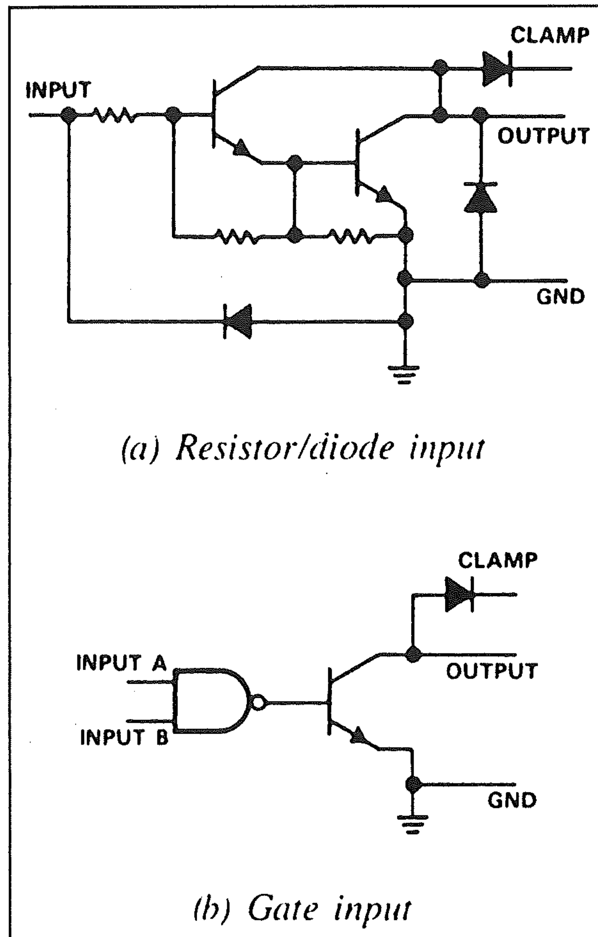
Aan digitale perifere drivers kan een aantal eisen (en combinaties daarvan) worden gesteld, zoals:

- maximaal vermogen;
- maximale spanning;
- maximale stroom;
- minimale snelheid.

Het maximaal vermogen

Digitale perifere drivers worden altijd toegepast in situaties waarin zij een bepaalde hoeveelheid vermogen moeten verwerken. Met de komst van de DMOS-schakelingen wordt een en ander wel gunstiger, maar in de praktijk is het niet mogelijk dat de driver zelf totaal geen vermogen dissipeert. De meeste plastic DIL-behuizingen zijn nog wel geschikt voor dissipaties tot 1 W, maar voor grotere vermogens zijn speciale behuizingen met koellichamen nodig.

6.16 Digitale perifere drivers



Figuur 3/6.16-2: Twee voorbeelden van de interne samenstelling van digitale perifere drivers:
a: weerstand/diode-ingang;
b: poort-ingang.

De maximale spanning

De drivers kunnen via de belasting (lamp, relais, enz.) op spanningen tussen 15 V en 100 V worden aangesloten. Ook deze spanningen worden altijd in de beschrijvingen vermeld.
Wanneer een driver voor schakeldoeleinden wordt gekozen moet extra aandacht worden besteed aan de maximale spanning. Een driver met een toelaatbare uitgangsspanning van 30 V hoeft bijvoorbeeld niet altijd geschikt te zijn om een 24 V relais aan te sturen, zelfs niet als de uitgang is voorzien van een clampdiode. De high-level uitgangsspanning na het schakelen (V_{OH}) bedraagt voor deze driver typisch 20 V. De zwaai van 24 V die de uitgang maakt zou in dit geval leiden tot secundaire doorslag en latch-up. Dit is een destructieve toestand die de driver zou kunnen beschadigen. Voor deze toepassing heeft de meest geschikte driver een V_{OH} van 30 V.
Perifere drivers die aan de ingang zijn voorzien van een interne besturingspoort hebben bovendien nog een extra 5 V voedingsspanning nodig.

De maximale stroom

Digitale perifere drivers kunnen meestal een uitgangsstroom tussen 100 mA en 2 A verwerken. Ook de benodigde stroom dient met zorg te worden gekozen. Van de meeste drivers wordt in de specificaties zowel de continue (gelijk)stroom als de toegelaten piekstroom vermeld.
Piekstromen worden gespecificeerd voor maximaal 10 ms en een duty-cycle (aan/uit-tijd) van 50 % of minder. Bedenk echter dat de vermelde piekstroom nooit mag worden overschreden, hoe kort de tijd ook of hoe klein de duty-cycle, omdat hierdoor "metaal-migratie" optreedt die vroeger of later tot defecten zal leiden.

De minimale snelheid

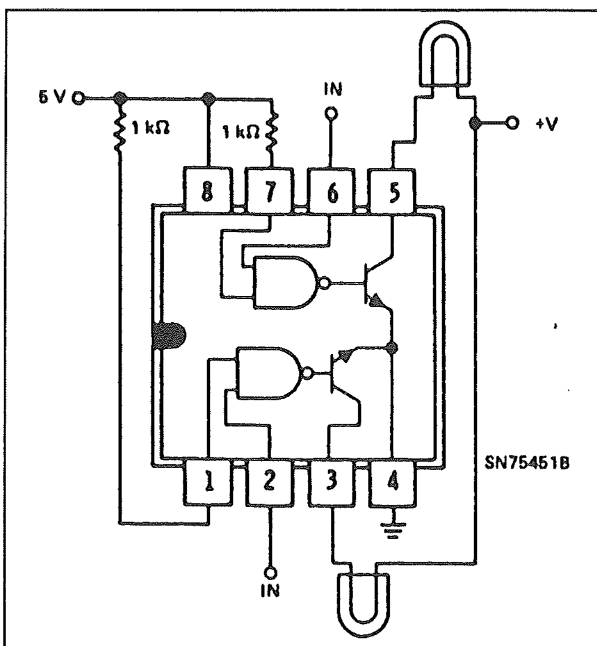
Vaak worden digitale perifere drivers als schakelaars gebruikt, waardoor ze lang in dezelfde toestand zullen blijven of met een zeer lage frequentie van stand veranderen.
Maar bij andere toepassingen, bijvoorbeeld als clockdriver voor geheugens, wordt een hogere snelheid (bijvoorbeeld 10 MHz) gevraagd.

Bij lage snelheden en DC mogen de grenzen van vermogen, spanning en stroom nooit worden overschreden. Bij bedrijf met zeer hoge snelheden kan intern echter een zeer grote extra dissipatie optreden. Dit wordt veroorzaakt door de (paracitaire) capaciteiten van de in de geïntegreerde schakeling gebruikte transistoren die typische stijg- en afvaltijden tot gevolg hebben.

Toepassingen

Inleiding

In het volgende paragraafjes worden enkele toepassingsvoorbeelden van digitale perifere drivers gegeven. De hierbij vermelde typenummers dienen ter illustratie en kunnen natuurlijk door andere worden vervangen. Let echter op de specificaties!



Figuur 3/6.16-3: Eenvoudige lampdriver zonder beveiligingen.

Het aansturen van gloeilampjes

In figuur 3/6.16-3 is een eenvoudige lamp-driver getekend, waarbij de gloeilampjes direkt worden aan- en uitgeschakeld met de stuursignalen op de (TTL)-ingangen. Er zijn geen beveiligingsmaatregelen getroffen en de uitgangstransistoren in de driver leiden de stromen naar aarde af.

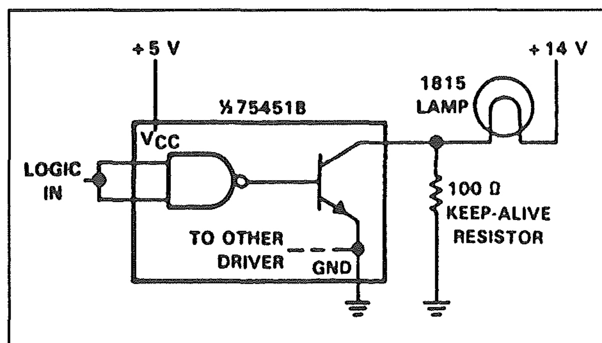
Gloeilampjes hebben echter een karakteristiek die de driver kan beschadigen. De weerstand van de gloeidraad verandert namelijk sterk met de temperatuur ervan. Een veel gebruikt lampje (type 1815) werkt bijvoorbeeld normaal bij 14 V en verbruikt dan 200 mA. Men zou denken dat een driver die 300 mA continu kan leveren met een piekstroom van 500 mA hiervoor wel geschikt is. Gloeilampjes hebben echter een inschakelstroom vanuit de koude toestand die ongeveer tien maal zo groot is als de bedrijfsstroom. Bij de 1815 werd 2,7 A gemeten!

Men heeft dan twee mogelijkheden:

- men kiest een driver die geschikt is voor een piekstroom van 3 A;
- men beperkt de stroom tot een kleinere waarde.

Een methode om de stroomsterkte te beperken is een zogenaamde "keep-alive" weerstand toe te passen, zoals in figuur 3/6.16-4 getekend is. De 100 Ω weerstand die parallel aan de uitgangstransistor is geschakeld neemt bij uitgeschakelde transistor ongeveer de helft van de nominale stroom op, waardoor de lichtopbrengst daalt tot circa 10 % van de normale waarde. De gloeidraad van het lampje wordt hierdoor reeds behoorlijk opgewarmd. Wanneer de lamp nu tot 100 % wordt ingeschakeld blijft de piekstroom beperkt tot 500 mA.

6.16 Digitale perifere drivers



Figuur 3/6.16-4: Stroombegrenzing met een "keep-alive" weerstand.

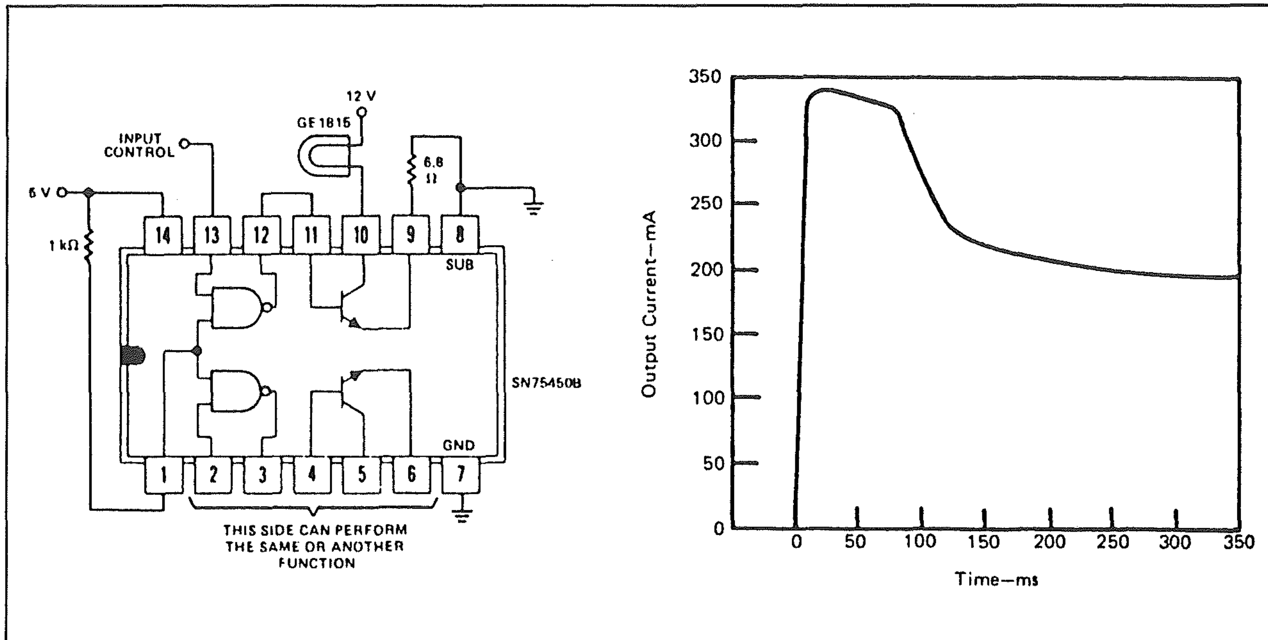
Nu is het voor controlelampjes niet aan te bevelen om ze "half" te laten branden. Men weet dan wel zeker dat ze niet kapot zijn, maar of de te controleren schakeling wel goed werkt is een andere vraag! Het is dan ook beter om de piekstroom op een andere manier te beperken. Er kan eenvoudig een begrenzingsweerstand in de emitterlijn van de uitgangstransistor worden opgenomen, zoals in figuur 3/6.16-5 wordt getoond.
Bij inschakelen zal de opwarmstroom proberen de maximale waarde te bereiken, maar wordt daarbij begrensd door de emitterweerstand. Is het lampje warm, dan heeft de gloeidraad een grotere weerstand en neemt de stroom verder af tot de nominale waarde.

Bij stroombegrenzing met een emitterweerstand wordt in deze weerstand natuurlijk energie verspild. Alle stroom blijft hier immers doorheen gaan. Bovendien zal de lamp dan niet op volle spanning branden. In figuur 3/6.16-6 is een andere manier voor stroombegrenzing zonder emitterweerstand te zien.
Wanneer de stroomversterkingsfactor h_{FE} van de transistor bekend is kan de uitgangsstroom worden begrensd door een basisweerstand te gebruiken. Is h_{FE} bijvoorbeeld 50 dan zal voor een collectorstroom van 250 mA de basisstroom 5 mA moeten zijn. De basisweerstand moet in dat geval 500 Ω zijn.

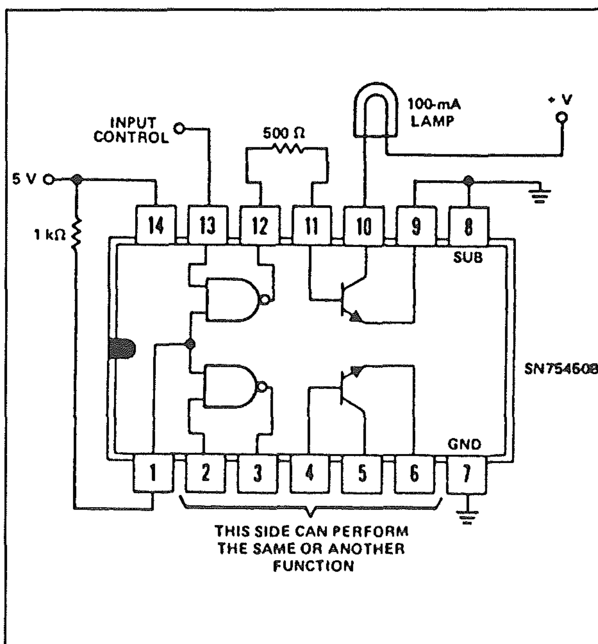
Het zal duidelijk zijn dat deze methode niet de beste is, omdat op verschillende belangrijke zaken geen invloed kan worden uitgeoefend.
De parameters h_{FE} en V_{BE} van de transistor kunnen per type nogal verschillen en zijn bovendien temperatuursafhankelijk.
Veel mooier is dan de schakeling uit figuur 3/6.16-7. Hierbij is de logische zijde van de schakeling voorzien van een tijdelement. In de begintoestand (lamp uit, logic control LAAG) is de condensator ontladen.
Gaat het stuursignaal HOOG, dan gaat de bovenste poort direct open zodat de bovenste transistor geleidt en de stroom door de lamp via de weerstand van 27 Ω naar aarde vloeit.
Na een korte tijd (in dit geval met 5,6 kΩ en 68 µF ongeveer 180 ms) is de condensator zover opgeladen dat ook de onderste poort open gaat. De onderste transistor gaat nu ook geleiden zodat de lampstroom de weg van de minste weerstand kiest en door de onderste transistor direct naar aarde vloeit.
De lamp werkt nu op de volle spanning en er gaat geen energie verloren in een begrenzingsweerstand.

Het aansturen van inductieve belastingen
Wanneer digitale perifere drivers worden gebruikt voor het aansturen van inductieve belastingen zoals relais en spoelen moet behalve de toegelaten stroom, spanning en vermogen ook de maximale schakelspanning in de gaten worden gehouden.

6.16 Digitale perifere drivers



Figuur 3/6.16-5: Stroombegrenzing bij het inschakelen van de gloeilamp met behulp van een emitterweerstand, met de bijbehorende stroom/spanning-karakteristiek bij het inschakelen.



Figuur 3/6.16-6: Stroombegrenzing met behulp van een basisweerstand.

Deze wordt bij de meeste drivers gespecificeerd. Dikwijls worden clamp-dioden toegepast om te voorkomen dat bij het uitschakelen van inductieve belastingen extreme spanningen op de uitgang van de driver komen (zie figuur 3/6.16-8). Deze hebben als enig nadeel dat bijvoorbeeld een relais iets minder snel afvalt.

Wanneer in de driver geen inwendige clampdioden aanwezig zijn of wanneer die niet in de belasting (bijvoorbeeld sommige DIL-reedrelais) zijn opgenomen, moeten ze extern worden aangebracht.

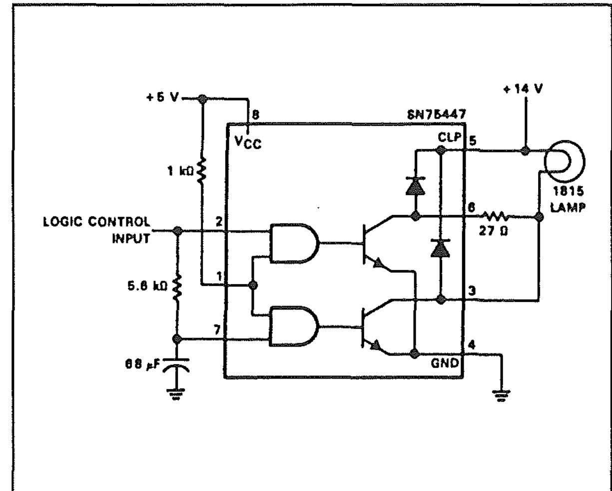
Ook in printers (en vroeger in bandponsers) worden inductieve belastingen gevonden: de zogenaamde "hamers". Bij "daisywheel"-printers worden hiermee complete karakters in één keer op het papier afgedrukt; bij matrix-printers bestaat een karakter uit verschillende puntjes die via stangetjes worden aangebracht. Kortom: er wordt gebruik gemaakt van hef-, trek-, duw-, zuig- of schuifmagneten die een elektrisch signaal omzetten in een mechanische beweging. In het geval van een afdrukmechanisme ligt het voor de

6.16 Digitale perifere drivers

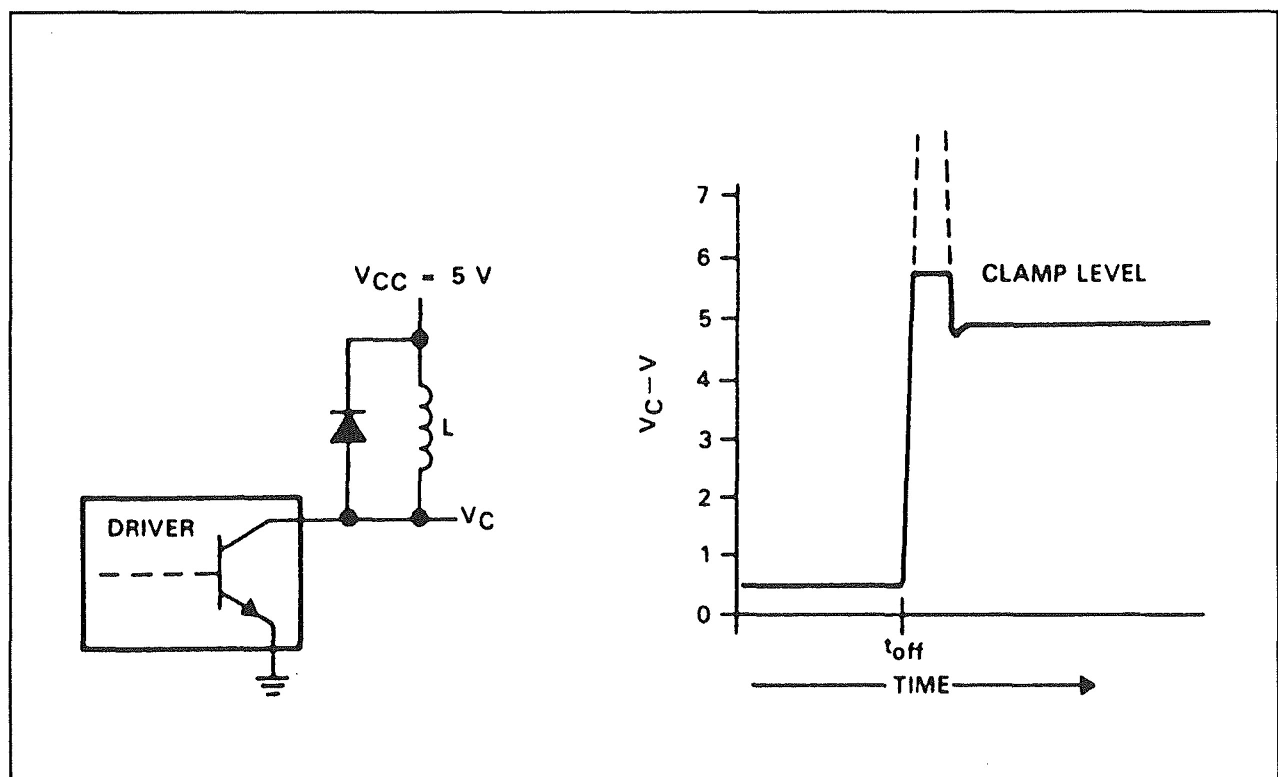
hand dat meerdere spoeltjes tegelijk bekrachtigd kunnen zijn (figuur 3/6.16-9). Er moet dan rekening worden gehouden dat de driver het gezamenlijke vermogen (onder de slechtste condities) kan dissiperen.

Het aansturen van motoren

Ook voor het aansturen van motoren zijn digitale perifere drivers zeer geschikt. Maar in de meeste gevallen verbruiken motoren zoveel stroom, dat zelfs de zwaarste digitale drivers niet in staat zijn de belasting rechtstreeks te sturen. Tussen de motor en de driver zal men in de meeste gevallen een extra transistor zetten en het zal logisch zijn dat POWERFET's, vanwege hun zeer lage inwendige weerstand, daar uitermate geschikt voor zijn.

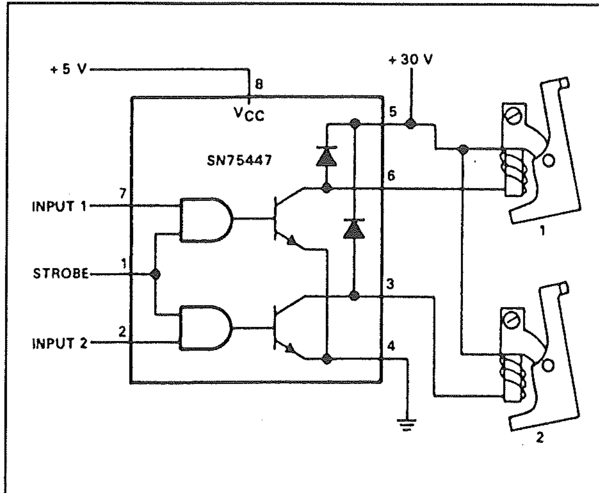


Figuur 3/6.16-7: Ideale lamp-driver met tweetraps begrenzingsschakeling.



Figuur 3/6.16-8: Aansturing van een inductieve belasting (bijvoorbeeld een relais) met diode-clamping van de uitgangsspanning.

6.16 Digitale perifere drivers

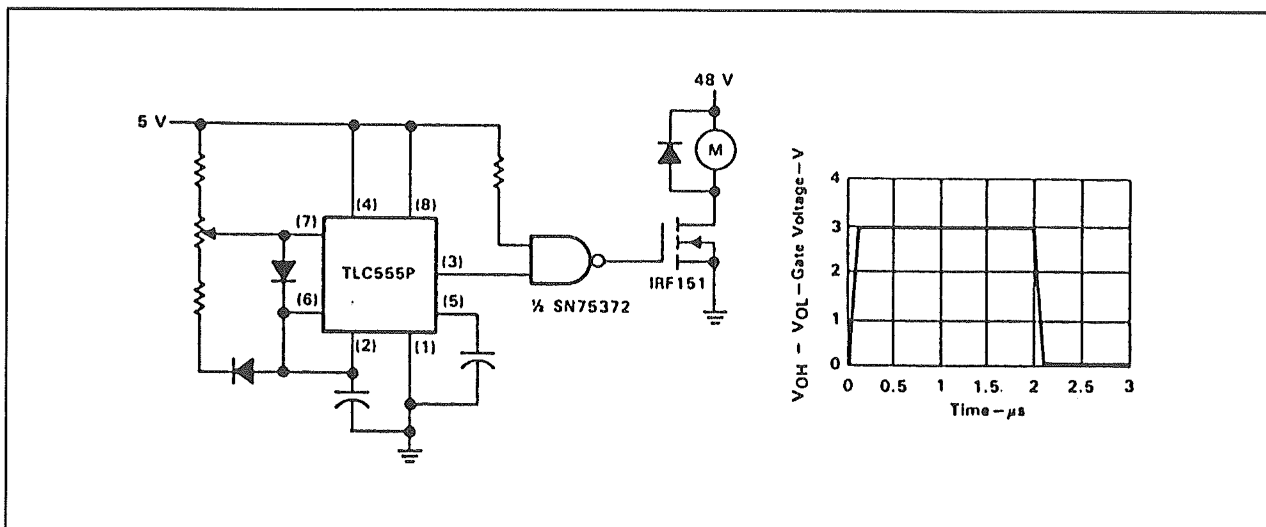


Figuur 3/6.16-9: Een voorbeeld van een dubbele hamer-driver met een digitale perifere driver met ingebouwde clamp-dioden.

Als voorbeeld wordt in figuur 3/6.16-10 een schema getekend van een motorregeling, waarbij de motor wordt aangestuurd door middel van een blokgolf met instelbare aan/uit-verhouding. Deze blokgolf wordt gegenereerd door een timer-IC van het type 555. De TLC-uitvoering daarvan is uiteraard niet in staat voldoende vermogen te leveren voor het aansturen van de zware MOSFET IRF151. De uitgang van de timer stuurt dus een digitale perifere driver van het type SN75372 en de uitgang van de driver stuurt de gate van de MOSFET. Let ook nu op de clamp-diode over de motorwikkeling, die noodzakelijk is vanwege de pulsvormige aansturing van de motor.

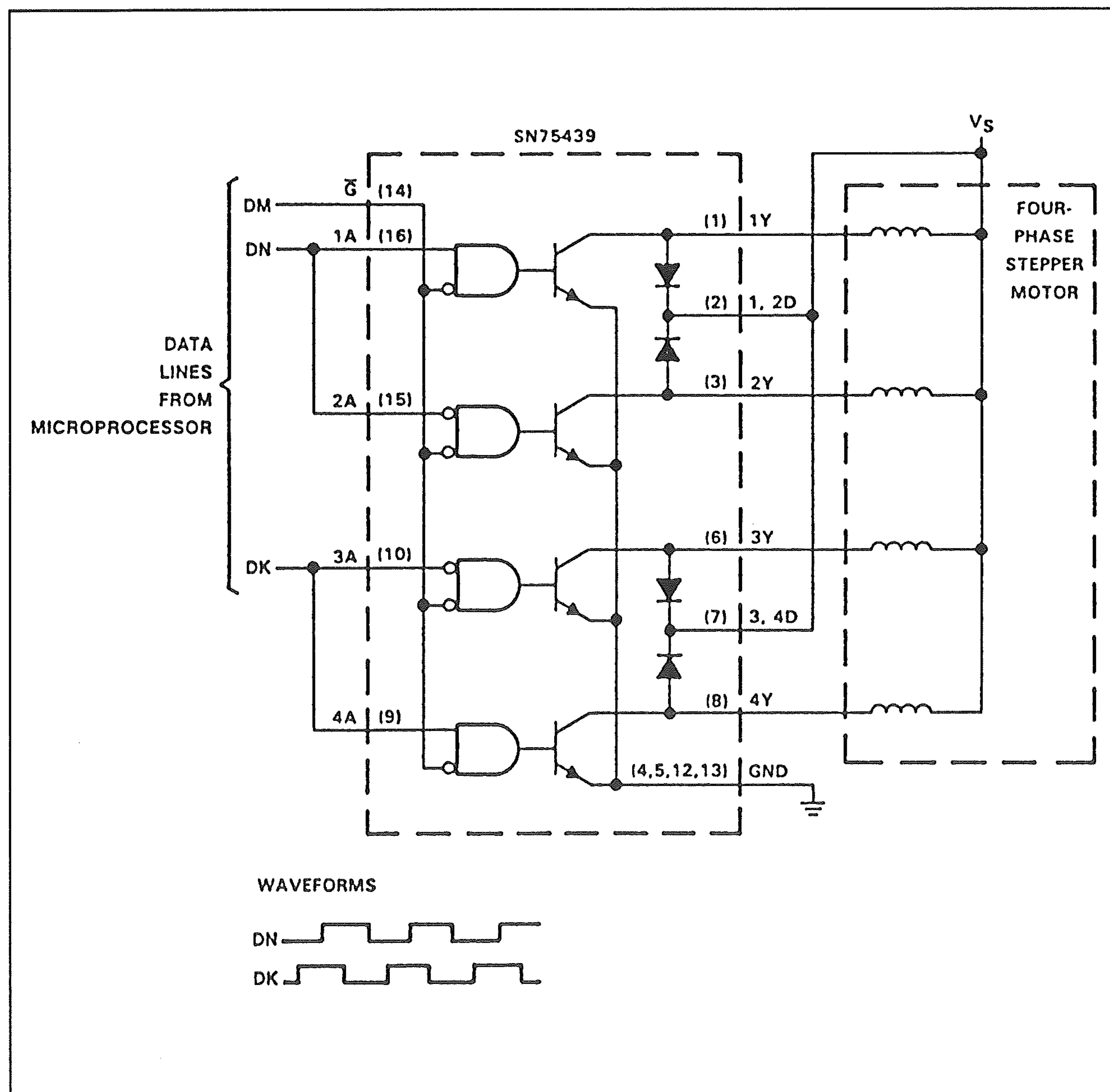
Digitale perifere drivers zijn ook zeer geschikt voor het aansturen van stappenmotoren.
Dit type motor is immers voorzien van verschillende spoelen, waardoor in een bepaalde onderlinge tijdsrelatie korte impulsvormige stroomstoten gestuurd moeten worden. Er zijn diverse digitale perifere drivers ontwikkeld voor dit soort toepassingen. Als voorbeeld wordt in figuur 3/6.16-11 een schema getekend rond de SN75439.

Diverse overige toepassingen
Behalve voor de hier genoemde toepassingen zijn digitale perifere drivers voor vele andere bruikbaar.
Te denken valt aan drivers voor magnetische kern-geheugens, display's, schakelende voedingen, enzovoorts.



Figuur 3/6.16-10: Het aansturen van een motor met een digitale perifere driver.

6.16 Digitale perifere drivers

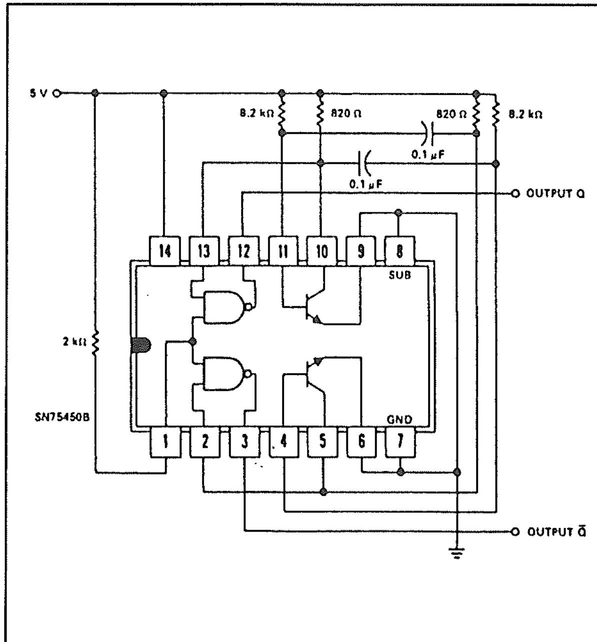


Figuur 3/6.16-11: Een SN75439 is ontwikkeld voor het aansturen van stappen-motoren.

Verder kunnen ze worden gebruikt als niveau-verschuivers tussen verschillende soorten logische families.
In het geval dat alle aansluitingen van de transistoren naar buiten worden uitgevoerd zijn deze natuurlijk ook als ingang te gebruiken.
Hierop kunnen bijvoorbeeld sensoren worden aangesloten.

Als voorbeeld van een niet alledaagse toepassing wordt in figuur 3/6.16-12 een blokgolf oscillator voorgesteld, opgebouwd met twee drivers. De tijdbepalende elementen zijn tussen de transistoren van de drivers aangesloten waardoor een blokgolf-oscillator met digitale uitgang (TTL-niveau) ontstaat, die in staat is vrij veel vermogen te leveren.

6.16 Digitale perifere drivers



Figuur 3/6.16-12: Een digitale perifere driver toegepast als blokgolf-generator.

3/6.17

Toepassen van geheugen-modulen in computers

Inleiding

De honger naar geheugen

Sinds zo'n 10 jaar geleden de eerste personal computers (PC's) verschenen is daar een enorme hoeveelheid software voor geschreven.
Het is echter een feit dat hoe "gebruiksvriendelijker" de programma's worden, hoe meer geheugen ervoor nodig is. Het 640 kB werkgeheugen waar de eerste PC's mee werden uitgerust (bijvoorbeeld het XT-type) is tegenwoordig voor veel software, denk maar aan Windows, niet meer genoeg.
Wie nu een PC aanschaft dient er rekening mee te houden dat 4 MB in de richting komt, maar dat uitbreiding tot minstens 8 MB mogelijk moet zijn. Denk hierbij maar aan besturings-systemen als OS/2 en New Technology!

Geheugen-modulen

De PC-fabrikanten zagen dit snel in en voorzien de PC's nu van speciale geheugen-sockets.
In plaats van IC's worden hier verwisselbare geheugen-modulen in geplaatst. Dank zij deze sockets voor geheugen-modulen is men zélf vrij eenvoudig in staat het geheugen van een PC uit te breiden tot boven de standaard 2 MB die nu in de meeste gevallen wordt geleverd.

Wat voor geheugen?

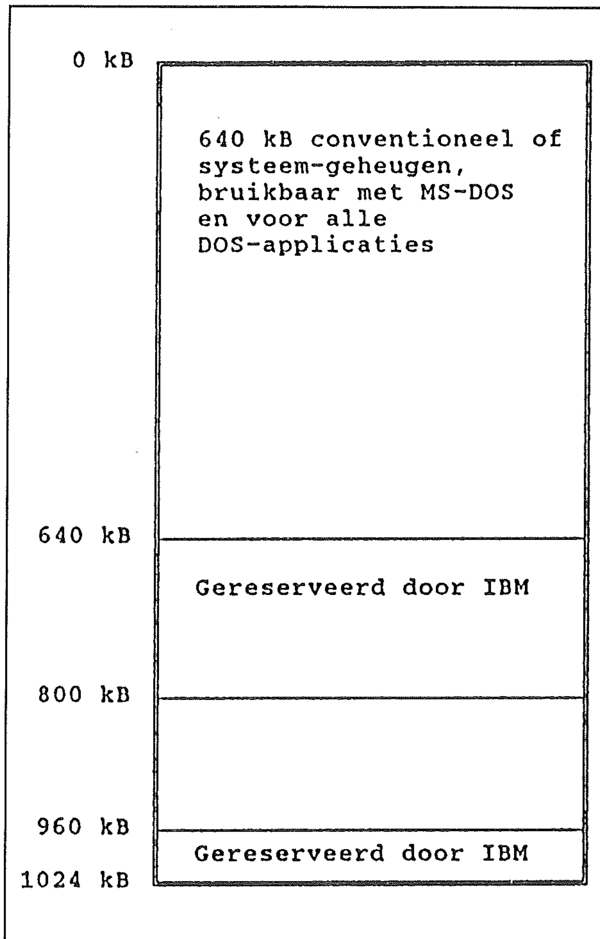
Maar alvorens dieper in te gaan op de samenstelling en de toepassing van deze geheugen-modulen is het noodzakelijk enige toelichting te geven op de indeling van het geheugen van een PC. Er worden op dit gebied tegenwoordig nogal wat termen gebruikt, die misschien niet voor iedere "gewone" elektronicus even duidelijk zijn.
Deze termen zijn:
- standaard geheugen;
- expanded geheugen;
- extended geheugen;
- cache-geheugen.

Het geheugen van de PC

Inleiding

Over de geheugenstructuur van 80286-, 80386- en 80486-systemen bestaan tal van misverstanden. In de meeste gevallen worden dergelijke computers tegenwoordig geleverd met 2 MB RAM-geheugen.
Algemeen bekend is dat MS-DOS slechts 640 kB geheugen ondersteunt, de beruchte "DOS-barriére". Als dat het geval is, wat heeft men dan aan die extra 1,34 MB geheugen in de computer? Hoe kan dit geheugen nuttig ingezet worden? Wat is het verschil tussen extended geheugen en expanded geheugen?

6.17 Toepassen van geheugen-modulen in computers



Figuur 3/6.17-1: De geheugenstructuur van de allereerste PC'c van het XT-type.

Het begin

De originele PC werd door IBM ontwikkeld rond de Intel 8088 microprocessor. Deze chip is in staat 1.024 kB geheugen rechtstreeks te adresseren.
Om niet meer te achterhalen redenen besloot IBM niet minder dan 384 kB van dit geheugen te reserveren voor speciale toepassingen, zoals het BIOS, videokaarten, harddisk controllers, etc.
Waarschijnlijk ging men er van uit dat toepassingsprogramma's toch nooit meer dan enige 100 kB aan geheugen nodig hadden.
Bedenk dat op dat moment alleen CPM-computers voor de grote markt bestonden waarbij 64 kB geheugen al heel wat was!

Geheugenstructuur van de PC

Het gevolg van deze beslissing was dat er slechts 1.024 kB - 384 kB = 640 kB overbleef voor de gebruiker.
MicroSoft hield daar bij de ontwikkeling van het besturingssysteem MS-DOS uiteraard rekening mee en vandaar dat MS-DOS slechts deze onderste 640 kB van het geheugen kan gebruiken.

De geheugenstructuur van een originele PC is getekend in figuur 3/6.17-1.

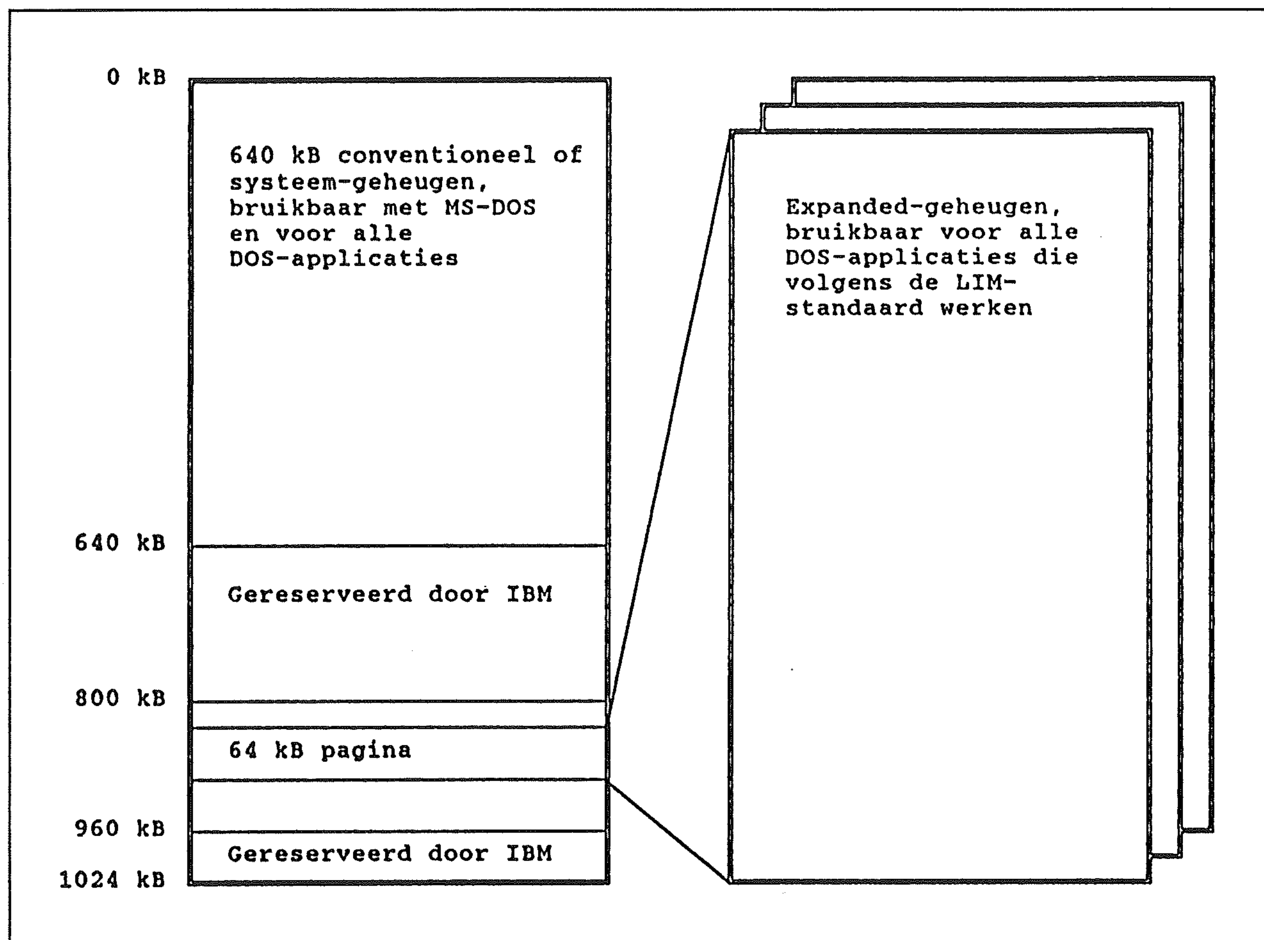
Geheugengebrek

Door het toenemen van de complexiteit van gebruikersprogramma's en van de databestanden die gebruikers nodig hadden bleek snel dat deze 640 kB barriére een probleem werd. Met name gebruikers van Lotus 1-2-3 zagen deze grens als een ernstige beperking bij het werken met hun programma. Vandaar dat er gesprekken tussen Lotus en Intel georganiseerd werden waarin naar oplossingen voor dit probleem werd gezocht.

Expanded Memory Specification (EMS)

Het gevolg hiervan was dat er tussen beide fabrikanten in 1985 een afspraak werd gemaakt, de zogenaamde EMS-standaard of Expanded Memory Specification.
Dat kwam er op neer dat Intel een hardware-uitbreiding ontwikkelde waardoor PC-gebruikers toegang konden krijgen tot 8 MB extra geheugen. Dat geheugen werd "expanded memory" genoemd.
Het principe is in feite erg eenvoudig en berust op wat men "bankswitching" noemt.

6.17 Toepassen van geheugen-modulen in computers



Figuur 3/6.17-2: Het principe van bankswitching tussen het systeem-geheugen van de PC en het expanded geheugen op het "Above Board".

Het extra geheugen is aangebracht op een insteekprint, die door Intel "Above Board" werd genoemd.
In het standaard geheugen van de PC wordt ergens in het gebied tussen de 800 en de 960 kB een zône van 64 kB gereserveerd. Dit blok of deze pagina wordt gebruikt als een soort venster, waarlangs men toegang krijgt tot blokken van 64 kB uit het "Above Board". De software zorgt ervoor dat razendsnel de gewenste 64 kB blokken uit het "Above Board" naar de bank in het standaard geheugen worden gecopieerd als men deze nodig heeft.
In figuur 3/6.17-2 is dit bankswitching proces grafisch toegelicht.

EMS-software
Lotus bracht een nieuwe versie van 1-2-3 uit die het EMS-geheugen ondersteunde. In een later stadium werd Symphony op de markt gebracht, een pakket dat ook van de mogelijkheden van het expanded geheugen gebruik kan maken.

De LIM-EMS V 3.2 standaard
Nog in datzelfde jaar 1985 sloot MicroSoft zich bij deze ontwikkeling aan en de EMS-standaard werd compatibel met de nieuwe versies van MS-DOS.
Vanwege deze samenwerking tussen Lotus, Intel en MicroSoft wordt de specificatie van het expanded geheugen sindsdien

6.17 Toepassen van geheugen-modulen in computers

LIM-EMS genoemd. De originele versie werd 3.2 genoemd. Deze LIM-EMS specificatie is een de facto standaard geworden. Tal van software-ontwikkelaars hebben pakketten op de markt gebracht die LIM-EMS geheugen ondersteunen! Om er enige op te sommen: 1-2-3, Framework, Excel-DOS, Paradox, SuperCalc, Symphony, Ventura-DOS, Smart en AutoCad.
Ook een heleboel grafische programma's, waar de behoefte aan veel geheugen voor de hand ligt, ondersteunen tegenwoordig LIM-EMS geheugen.

Enhanced Expanded Memory Specification
Het kon uiteraard niet uitblijven! Onder leiding van AST werd door een conglomeraat van hardware fabrikanten een concurrerend systeem op de markt gebracht. Dit werd Enhanced Expanded Memory Specification, kortweg EEMS, genoemd. Ook voor dit systeem heeft men uitbreidingskaarten nodig, de zogenaamde EEMS-kaarten.
In wezen ondersteunt EEMS alle mogelijkheden van LIM-EMS. Er is dus sprake van compatibiliteit en alle programma's die onder LIM-EMS draaien zullen dat ook doen onder EEMS.
EEMS biedt echter enige extra opties, die echter maar door weinig pakketten uitgebuit worden. Voor zover bekend in Concurrent DOS van Digital Research het enige besturingssysteem dat van deze extra mogelijkheden gebruikt maakt.

LIM-EMS V 4.0
In 1987 werd door de drie samenwerkende fabrikanten Intel, Lotus en MicroSoft een verbeterde versie van LIM-EMS op de markt gebracht, namelijk V 4.0.
De verbeteringen hebben in hoofdzaak betrekking op de volgende opties:

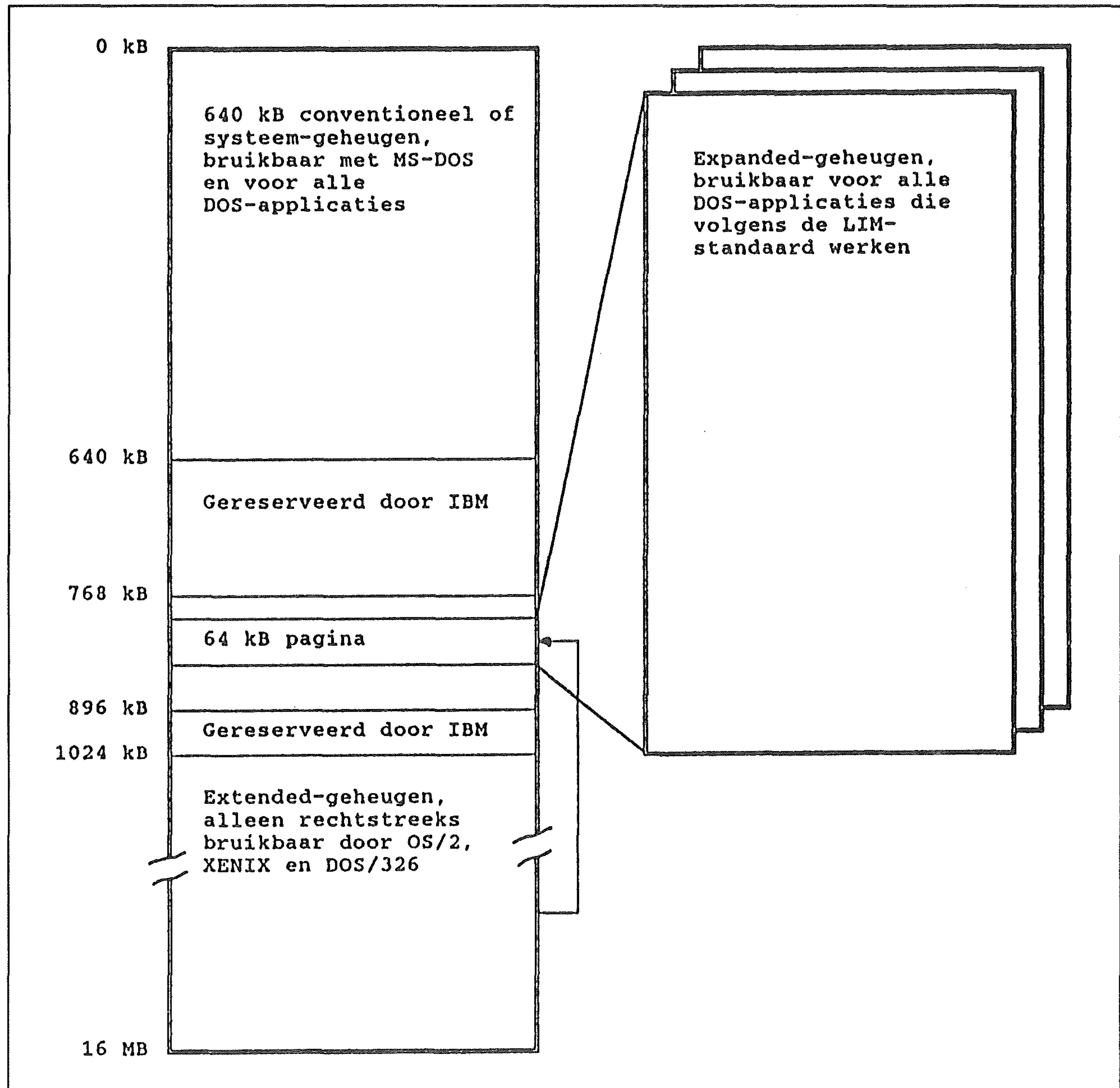
- met V 4.0 kan men niet minder dan 32 MB extra geheugen in de PC installeren;
- bij V 3.2 kon men alleen gegevens in het expanded geheugen onderbrengen, bij V 4.0 kan men echter ook programma's naar het EMS-geheugen laden;
- het is nu mogelijk meerdere pagina's van 64 kB in het systeem geheugen te installeren, zodat meer geheugen geswitched kan worden tussen systeem en expanded geheugen.

V 4.0 is geheel compatible met V 3.2, zodat alle oude programma's blijven werken. Er zijn echter sindsdien nieuwe versies van pakketten op de markt verschenen die alleen werken onder V 4.0 van LIM-EMS. Voorbeelden zijn Excel van MicroSoft en 1-2-3 van Lotus.

Algemene standaardisering
AST en de overige initiatiefnemers van de EEMS-standaard gingen na het uitkomen van V 4.0 van LIM-EMS overstag. Sindsdien kan men stellen dat LIM-EMS V 4.0 dé algemene standaard is wat betreft geheugen uitbreidingen voor de PC.
Tal van hardware- en softwareleveranciers leveren bij hun systemen of pakketten zogenaamde devicedrivers, waarmee men systeem en programma gebruik kan laten maken van expanded geheugen.

Extended geheugen
Ondertussen was er echter een geheel andere ontwikkeling door IBM in gang gezet. In 1983 bracht deze fabrikant de AT uit, een computer die niet meer rond de 8088 processor was gebouwd, maar rond Intel's nieuwe 80286.
Deze processor is in principe in staat 16 MB geheugen rechtstreeks te adresseren.

6.17 Toepassen van geheugen-modulen in computers



Figuur 3/6.17-3: De geheugenstructuur van een 80286-systeem.

De 15 MB boven het standaard PC-geheugen noemt men het "extended geheugen".
Maar omdat de compatibiliteit met de oude 8088 voorop stond heeft men de 80286 twee zogenaamde werkmodi gegeven:
- Real Mode;
- Protected Mode.

In de zogenoemde "Real Mode" werkt de 80286 in feite als een oude 8088. Er is dan maar 1.024 kB geheugen toegankelijk. Deze mode wordt gebruikt door alle MS-DOS applicaties die op een AT draaien.
In de zogenaamde "Protected Mode" kan de processor alle 16 MB aanspreken (in de veronderstelling dat de computer daarmee zou uitgerust zijn!). Deze mode

6.17 Toepassen van geheugen-modulen in computers

wordt echter alleen ondersteund door geavanceerde bedrijfssystemen zoals OS/2, New technology en UNIX.
De meeste AT's zijn tegenwoordig standaard uitgerust met 2 MB geheugen.
Er is dan dus 1 MB extended geheugen aanwezig, hoeveelheid die meestal tot 4, 8 of 16 MB is uit te breiden op de basisprint van het systeem.

De geheugenstructuur van een 80286-systeem is getekend in figuur 4/8.2-3.
Hieruit blijkt duidelijk dat men ofwel fysisch expanded geheugen moet inzetten of het extended geheugen als dusdanig moet initialiseren.

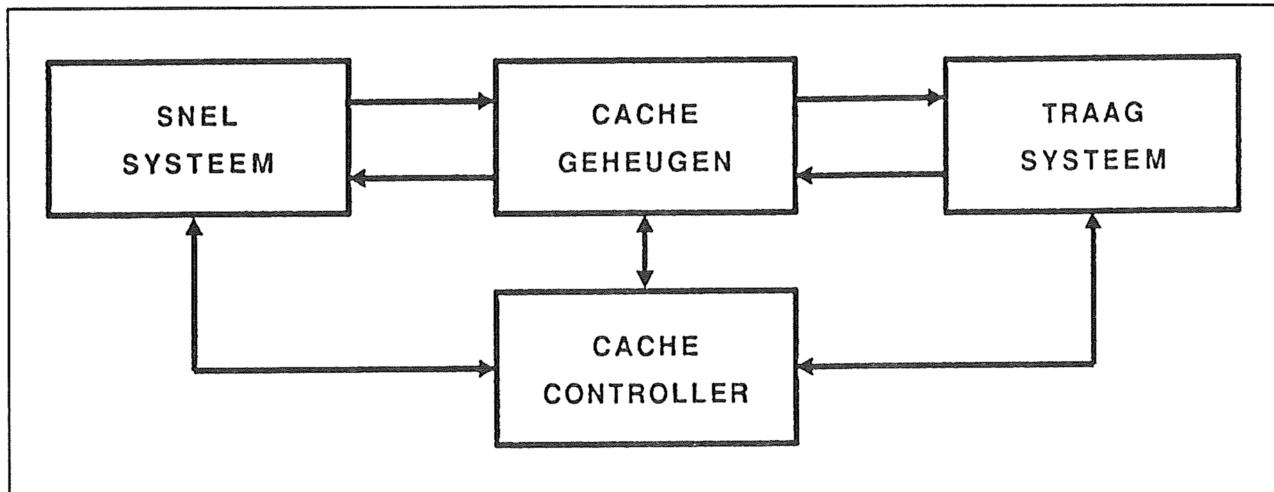
80386- en 80486-systemen
Hiervoor geldt in principe hetzelfde verhaal als voor de 80286. Ook deze processoren kennen verschillende modes, waaronder de Real en de Protected.
Wat geheugen betreft is het enige verschil met de 80286 dat deze processoren in principe in staat zijn niet minder dan 2^{32} verschillende geheugenadressen aan te spreken. Dat komt neer op ongeveer 4 GB! Er is dus in theorie geen grens aan de hoeveelheid extended geheugen die men in dergelijke systemen kan installeren. De dagelijkse praktijk is uiteraard de enige grens en meer dan 32 MB is op dit moment, zelfs in snelle systemen voor grafische toepassingen, niet noodzakelijk.

Cache-geheugen
Het ontwikkelen van cache-geheugen werd noodzakelijk toen de snelheid van de processoren steeds meer werd opgevoerd. Toen de op 16 MHz werkende 80286 door Intel op de markt werd gebracht bleek in de praktijk dat de snelheid van de standaard geheugenchips te laag was om de werksnelheid van deze processor te kunnen bijbenen. Het gevolg is dat er zogenaamde "waitstates" moeten worden ingevoerd.
De processor moet dan een of meerdere klokperioden wachten bij iedere toegang tot het systeem-geheugen om dit deel van de elektronica gelegenheid te geven de gevraagde gegevens aan te kunnen bieden respectievelijk te verwerken.
Met de introductie van op 33, 50 en zelfs 66 MHz werkende processoren nam dit probleem uiteraard alleen maar toe.

Men heeft dit probleem in eerste instantie opgelost door het systeem-geheugen op een bepaalde manier te organiseren. Een van de bekendste systemen is het zogenaamde "interleaved RAM". Daarbij wordt het geheugen ingedeeld in banken die ieder op zich afzonderlijk toegankelijk zijn voor de processor. Op deze manier kan men waitstates voorkomen door de processor opeenvolgende gegevens niet allemaal in één bank te laten schrijven, maar in opeenvolgende banken. Terwijl de eerste bank bezig is een gegeven te verwerken, kan de processor reeds een nieuw gegeven aan de tweede bank aanbieden enzoverder.
Het zal echter duidelijk zijn dat deze techniek alleen echt effectief is als de processor steeds opeenvolgende gegevens uit verschillende RAM-banken nodig heeft. In de praktijk zal dat niet altijd het geval zijn!

Vandaar dat men een systeem heeft ontwikkeld dat veel beter werkt, namelijk het invoegen in het systeem van een klein, zeer snel geheugen. Dit geheugen, dat 8 tot 64 kB groot kan zijn, is opgebouwd uit zeer snelle geheugenchips's. Het conventionele geheugen bestaat uit chip's met toegangstijden van 120 tot 100 ns.

6.17 Toepassen van geheugen-modulen in computers



Figuur 3/6.17-4: De positie van het cache-geheugen tussen de snelle processor en het trage systeemgeheugen.

Het kleine cache-geheugen kan echter werken met een toegangstijd van slechts 25 tot 45 ns. Dit geheugen werkt dus minstens drie maal sneller dan het conventionele geheugen.

Vaak ziet men tegenwoordig systemen aangeboden met 64 kB cache, waarbij echter vermeld wordt dat het mogelijk is de grootte van deze cache uit te breiden. Dat heeft echter geen zin! Het probleem is namelijk dat daardoor geen verdere snelheidswinst ontstaat.
Het cache-geheugen wordt bestuurd door een cache-controller, zie figuur 3/6.17-4.

Deze controller moet administreren welke gegevens er waar in de cache staan en waar deze gegevens in het conventionele geheugen thuis horen. Deze boekhouding is noodzakelijk omdat de cache uiteraard niets mag wijzigen aan de inhoudelijke samenstelling van het conventionele geheugen. Hoe groter nu de cache, hoe meer tijd de cache-controller nodig heeft om deze boekhouding te voeren. Deze controller-tijd beperkt dan weer de effectiviteit van het cache-systeem. Er bestaat dus een bepaalde grens in de grootte van de cache, waarbij verdere uitbouw van de cache geen enkele zin meer heeft. Deze grens is uiteraard afhankelijk van de snelheid waarmee de cache-controller de noodzakelijke administratie kan voeren. Op dit moment zijn alle specialisten het er over eens dat, met de huidige verwerkingssnelheid van de elektronica, 64 kB die grens is.

Er zijn verschillende technieken ontwikkeld om het cache-geheugen zo efficiënt mogelijk te benutten:
- Fully associative cache;
- Set associative cache;
- Direct mapped cache;
- Write through cache;
- Write back cache.

Deze worden nu in het kort besproken.
- **Fully associative cache**
  Bij deze techniek bewaart de cache de gegevens van de laatste N adressen van het conventionele geheugen, waarnaar de processor geschreven of waaruit hij gelezen heeft. N is dan gelijk aan de grootte van de cache. Basisgedachte is dat het in de praktijk meestal zo is dat

de processor vaak achter elkaar met identieke geheugenadressen moet samenwerken.

- **Set associative cache**
  Bij deze techniek wordt het cache-geheugen in twee of vier blokken verdeeld, die ieder worden toegekend aan een deel van het conventionele geheugen. Men spreekt dan van een "two-way set" of van een "four-way set". Voordeel van deze techniek is dat de cache-controller nu slechts een deel van het cache-geheugen per keer moet administreren, waardoor de snelheid verhoogd wordt.
- **Direct mapped cache**
  Bij de "direct mapped cache" wordt aan ieder adres van de cache een aantal adressen van het conventionele geheugen gekoppeld.
  Dit is dus een soort van banking-techniek. Deze koppeling is star en wordt gemaakt door middel van de eerste 10 tot 14 bits van het adres. De cache-controller kan aan deze bits zien met welk deel van het conventionele geheugen de processor wil samenwerken en vult het adres van de cache met de noodzakelijke gegevens. Dit vereist echter een zeer snelle en zeer intelligente cache-controller! Als deze gegevens reeds in de cache aanwezig zijn kunnen deze zeer snel door de processor worden uitgelezen.
- **Write through cache**
  Bij deze techniek worden alle gegevens door de processor rechtstreeks naar het conventionele geheugen geschreven, maar ook naar de cache. Deze techniek werkt zeer snel, maar is alleen effectief als de processor deze gegevens weer snel na het schrijven opnieuw nodig heeft.
- **Write back cache**
  Bij deze techniek schrijft de processor alleen naar de cache. Het conventionele geheugen wordt alleen maar geupdated als de cache-controller vaststelt dat data op geheugenadressen door de processor is gewijzigd.

De effectiviteit van deze cache-technieken is afhankelijk van de grootte van het cache-geheugen, de toegepaste techniek en het type cache-controller. In de tabel van figuur 3/6.17-5 worden enige systemen met elkaar vergeleken. Deze gegevens zijn afkomstig van Intel en hebben betrekking op een 80386-systeem. Uit de "performance factor" blijkt inderdaad duidelijk dat 64 kB cache de grens is. Bij een groter cache-geheugen neemt de snelheidswinst zelfs weer iets af! Ook blijkt duidelijk dat 8 kB de onderste grens is. Een kleiner cache-geheugen werkt weer vertragend op het systeem.
De effectiviteit van een systeem wordt dus door een externe processor-cache met 25 tot 40 % verbeterd.

Parity Error technieken

Inleiding
De meeste geheugen-modulen hebben een indeling in woorden van 9 bit breed. Dit is in eerste instantie verbazingwekkend. Gegevens worden in een computer, zo meent iedereen, toch verwerkt als bytes die een breedte hebben van 8 bit. Toch is dat niet het geval wat betreft het systeemgeheugen!
Om dit te verklaren moet even dieper worden ingegaan op een van de meest onbekende elektronische technieken, die IBM in de Personal Computer heeft ingebouwd: Parity Error Checking.

6.17 Toepassen van geheugen-modulen in computers

Cache Type	Cache Size	Location Size(bytes)	Hit Rate	Performance Factor
Direct-mapped	1K	4	41%	0.91
Direct-mapped	8K	4	73%	1.25
Direct-mapped	16K	4	81%	1.35
Direct-mapped	32K	4	86%	1.38
Direct-mapped	32K	8	91%	1.41
Direct-mapped	64K	4	88%	1.39
Direct-mapped	64K	8	92%	1.42
Direct-mapped	128K	4	89%	1.39
Direct-mapped	128K	8	93%	1.42
Two-way set	32K	4	87%	1.39
Two-way set	64K	4	89%	1.40
Two-way set	64K	8	93%	1.42
Two-way set	128K	4	89%	1.40

Figuur 3/6.17-5: Een tabel van Intel, waaruit de effectiviteit van diverse groottes van cache-geheugens wordt vergeleken.

Parity error
Iedere PC-gebruiker of -gebruikster zal wel eens ooit meegemaakt hebben dat het systeem vastloopt met een foutmelding:

Parity error
System halted

Men moet dan opnieuw opstarten en in de meeste gevallen is er verder niets aan de hand.
Die "Parity error" is een van de minst bekende en meest onbegrepen foutmeldingen die het systeem kan geven. En dat komt omdat men er, als alles goed is, maar zelden mee te maken krijgt.

Controle op data-integriteit noodzakelijk
De man of vrouw voor het beeldscherm heeft, onbewust, een in feite ongelooflijk groot vertrouwen in de verzameling elektronische onderdelen waaruit een PC is samengesteld. Want die elektronische onderdelen moeten er voor zorgen dat gegevens, die via het toetsenbord worden ingevoerd, ook als dusdanig in het geheugen van het systeem terecht komen en vanuit dit geheugen worden overgebracht naar de harde schijf of naar een floppy.
Nu is dát nog niet zo'n probleem. Maar men moet er ook zeker van zijn dat die eenmalig in het systeem-geheugen ingevoerde gegevens in de loop der uren niet veranderen.
Met andere woorden: als men om negen uur 's ochtends een spreadsheet vol getallen in het geheugen intikt, dan moet men er zeker van zijn dat die gegevens om 17h00 nog steeds ongewijzigd in dat systeem-geheugen staan en bij het saven van het spreadsheet foutloos op de harde schijf terecht komen.

6.17 Toepassen van geheugen-modulen in computers

En die eis is voor de elektronica heel wat moeilijker te realiseren!

Dynamische RAM's

Alle PC-geheugens zijn namelijk uitgevoerd onder de vorm van dynamische RAM's. De gegevens, elektrische spanningen, worden in deze IC's opgeslagen in een minuscuul condensatortje van minder dan 1 pF.
De spanning over zo'n kleine condensator zal echter snel weglekken met als gevolg dat alle gegevens verloren zouden gaan als daar niet "iets" op gevonden was. Dat "iets" heet "refresh". De inhoud van het gehele systeem-geheugen wordt vele malen per seconde volledig ververst. Een ingewikkeld elektronisch proces, dat volledig bestuurd wordt door de processor.
En het is voldoende dat er bij deze elektronische bewerkingen iets mis gaat met de voedingsspanning van het systeem, om gegevens in het geheugen te verminken.
Het is bijvoorbeeld voldoende dat op een ongelukkig moment ergens in huis een koelkast inschakelt.
Bij het inschakelen van een motor ontstaan stoorspanningen op het 220 V wisselspanningsnet. Deze smalle stoorpieken dringen door tot de systeemvoeding van de PC. Hoewel alle PC-voedingen voorzien zijn van uitgebreide onstoringsnetwerken kan het toch gebeuren dat zo'n smalle stoorpuls gedeeltelijk doordringt tot de +5 V voedingsspanning van de dynamische RAM's. En dan kan deze stoorpiek tot gevolg hebben dat de inhoud van één of meerdere geheugencellen verloren gaat.

Een IBM-idee

De aartsvader van de PC, IBM, heeft dit probleem onderkend en maatregelen in het systeem ingebouwd die er voor moeten zorgen dat de gegevens-integriteit van de in het systeem-geheugen opgeborgen gegevens gecontroleerd wordt.
De techniek die daarvoor gebruikt wordt, wordt "Parity Error Checking" genoemd. Deze techniek is nadien (uiteraard) door alle cloon-fabrikanten overgenomen en is nu standaard in ieder systeem ingebouwd.

Negen bit per gegeven

Zoals algemeen bekend worden gegevens in de PC opgeslagen onder de vorm van bytes. Eén byte bestaat uit acht bit. Ieder bit kent slechts twee toestanden, namelijk "L" of "H".
In tegenstelling tot wat algemeen wordt aangenomen, worden deze 8 bit brede bytes achter niet in 8, maar in 9 geheugencellen opgenomen. De negende geheugencel wordt gebruikt als zogenoemd "parity-bit" en de inhoud van dit bit wordt gebruikt om de data-integriteit van de byte te controleren.

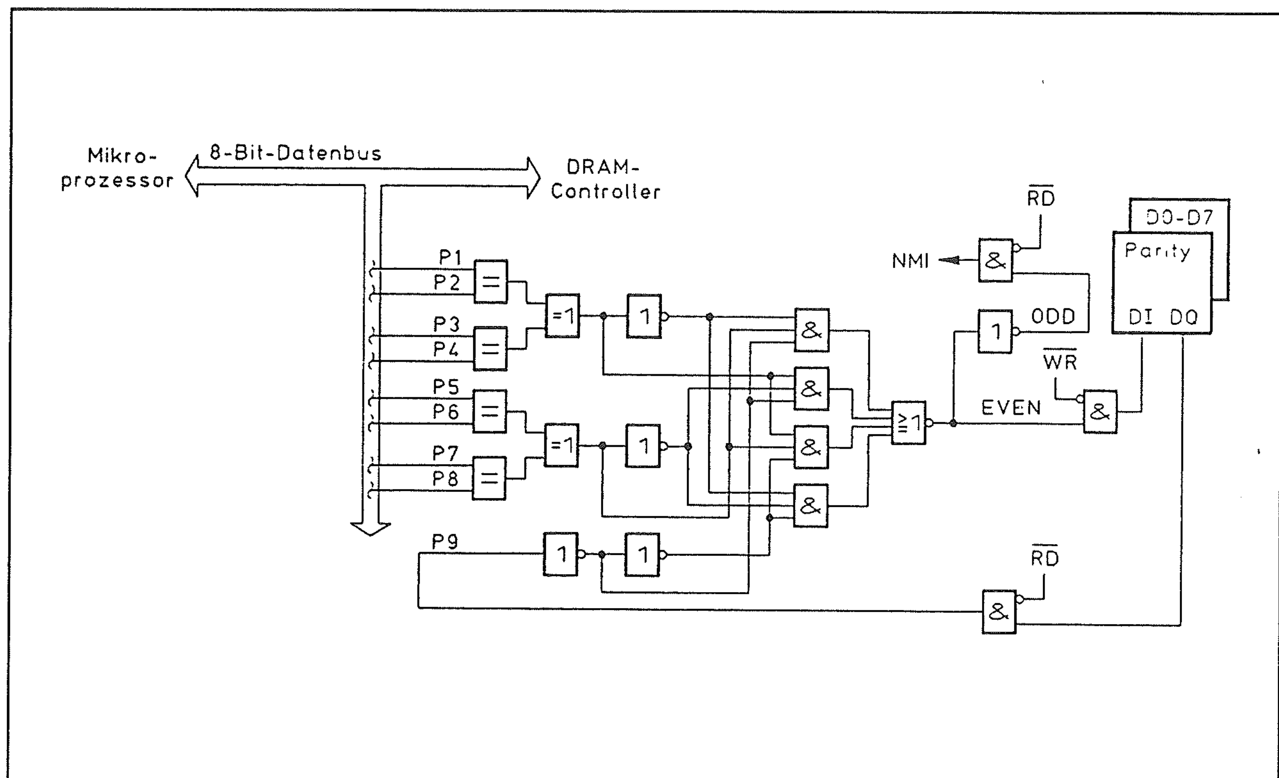
De codering van het parity-bit

IBM heeft een zeer eenvoudig systeem bedacht voor het controleren van de integriteit van een byte.
Het systeem telt het aantal hoge bits in de byte. Is dit aantal even, dan wordt het parity-bit "L" gemaakt.
Zijn er een oneven aantal "H"-bits in de byte, dan wordt het negende pariteits-bit "H" gemaakt.
Het byte wordt nu, samen met het pariteits-bit in het systeem-geheugen opgeslagen.
Deze techniek wordt toegelicht in de tabel van figuur 3/6.17-6. In het bovenste voorbeeld bestaat het byte uit vier hoge bits en wordt het parity-bit "L". In het onderste voorbeeld bestaat het byte uit vijf hoge bits, met als gevolg dat het parity-bit "H" wordt.

6.17 Toepassen van geheugen-modulen in computers

DATA-BITS								AANTAL	PARITY-BIT
0	1	2	3	4	5	6	7	HOOG	
L	L	H	H	L	H	L	H	VIER	L
H	H	L	H	L	L	H	H	VIJF	H

Figuur 3/6.17-6: De codering van het parity-bit toegelicht aan de hand van twee voorbeelden.



Figuur 3/6.17-7: Een praktische schakeling voor het genereren van het parity-bit.

De controle op de pariteit

Door de gekozen codering van het parity-bit zal een byte + parity-bit steeds een even aantal hoge bits bevatten! Bij het uit het systeem-geheugen halen van gegevens kan dit feit op een tamelijk eenvoudige manier elektronisch gecontroleerd worden.Leest het systeem een byte plus parity-bit uit het systeem-geheugen en stelt de elektronica vast dat er in die 9 bits een oneven aantal "H"-en zit, dan wordt dit doorgegeven aan het systeem en weet dit dat er iets fout is gegaan met de in het geheugen opgeslagen byte.

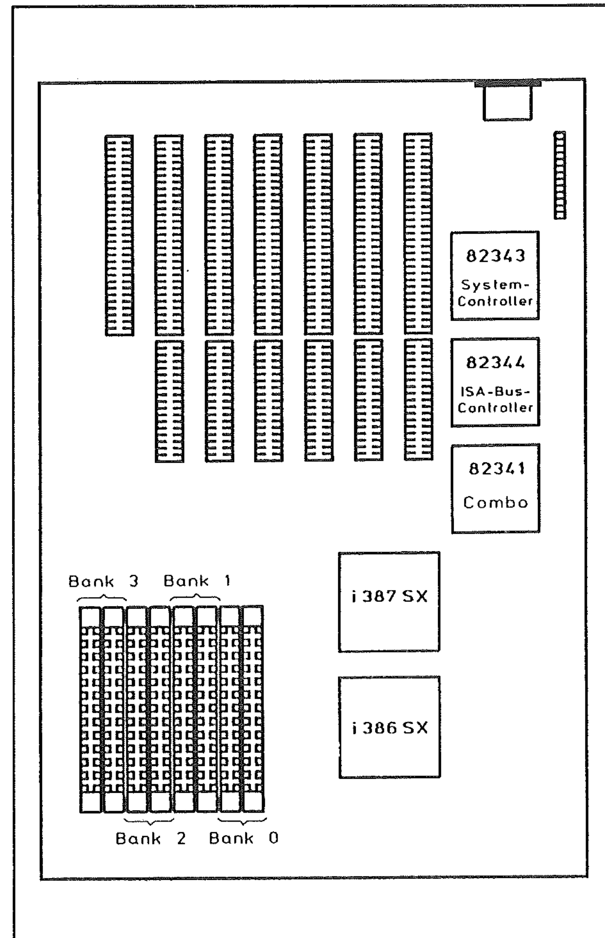
Op de hoofdprint van de PC is een speciaal IC aanwezig, dat samen met de RAM-controller zorgt voor het coderen en nadien weer decoderen van de parity-bit.

Als een parity-fout wordt vastgesteld vraagt dit IC een zogenoemde "Non Mas-

kable Interrupt" (NMI) aan bij de processor.
De interne structuur van de processor zorgt ervoor dat het systeem onmiddellijk stopt met alle werkzaamheden en met absolute prioriteit een bepaalde routine uit het BIOS gaat uitvoeren. Deze routine zet de foutmelding op het scherm en zet de processor nadien in een oneindig te doorlopen lus.
Het gevolg is dat het systeem "hangt" en er niets anders op zit dan te herstarten.

Het genereren van het parity-bit

Voor het genereren van het negende parity-bit wordt in de meeste gevallen een eenvoudige schakeling gebruikt, opgebouwd uit een aantal poorten. In figuur 3/6.17-7 is een voorbeeld van een praktische schakeling getekend.



Figuur 3/6.17-8: Speciale "slot's" voor geheugenmodulen op de moederprint van een moderne PC met een 386-processor en een 82343 system-controller.

Geheugen-modulen in de PC

Inleiding

Naast de aloude "slot's" voor uitbreidingsprinten, zoals modems, scanners, geluidskaarten, etc, heeft iedere moderne PC tegenwoordig ook een aantal "slot's" voor geheugen-modulen. Deze "slot's" zijn steeds aanwezig onder de vorm van een aantal "banken". Moderne moederprints beschikken over vier banken, waarin telkens twee geheugen-modulen gestoken kunnen worden. Op deze manier is het mogelijk in totaal 8 x 4 = 32 MB aan geheugen in het systeem aan te brengen.

Want er zijn tegenwoordig inderdaad geheugen-modulen op de markt, met een capaciteit van 4 MB.

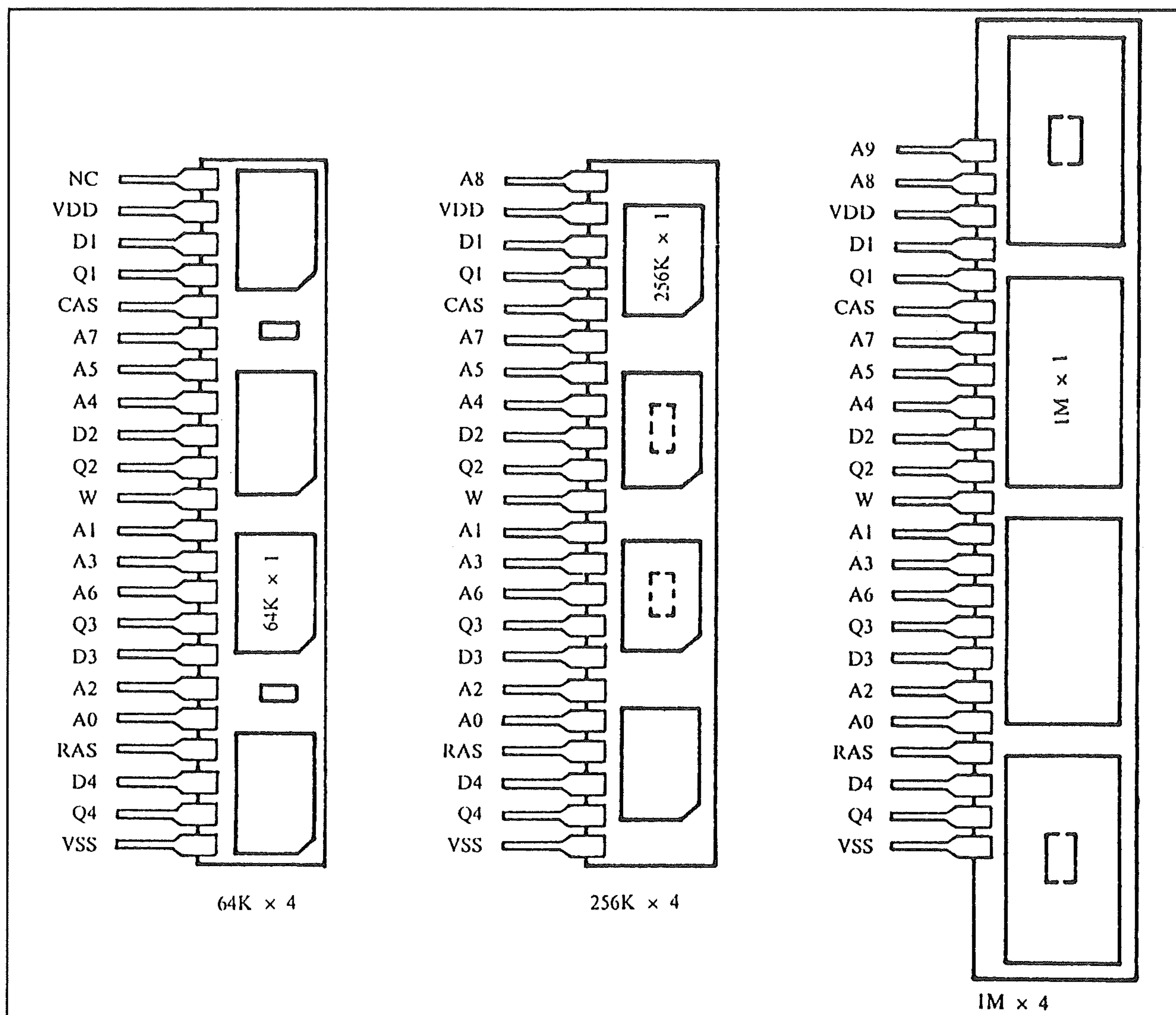
SIP contra SIMM

Let op dat er twee soorten geheugenmodulen zijn: de van insteekcontacten voorziene SIP's (Single-In-Line-Packages) en de penloze socket-typen SIMM's. Het zijn beide kleine printjes waarop een aantal geheugen-IC's is gemonteerd.

SIP's

De eerste geheugen-modulen die werden ontwikkeld waren van het SIP-type.

6.17 Toepassen van geheugen-modulen in computers

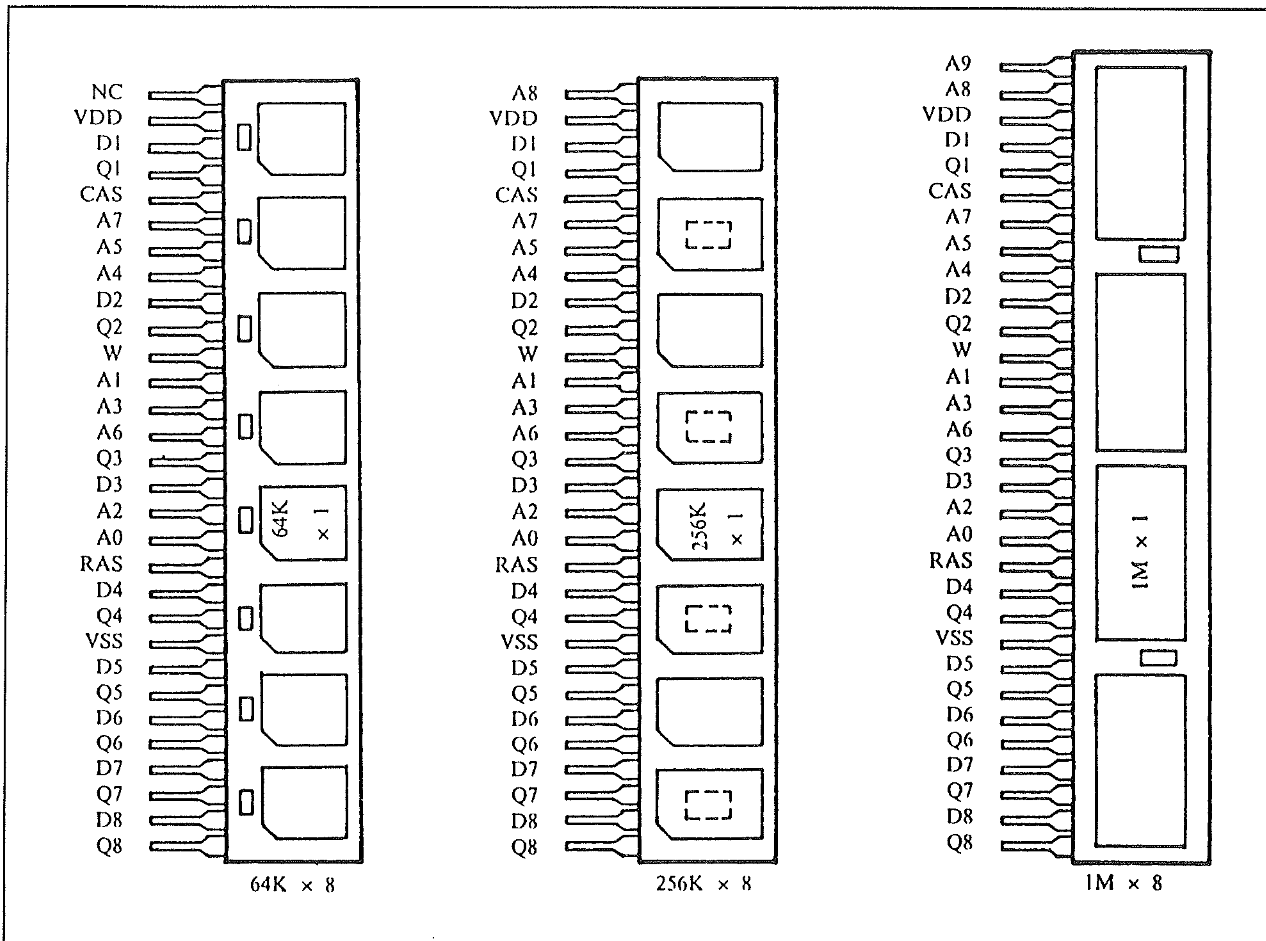


Figuur 3/6.17-9: Vergelijking van vier bit brede SIP's. Voor de aansluitpennen is steeds dezelfde volgorde aangehouden.

Er is al vanaf het begin naar gestreefd om de afmetingen en de aansluitpennen te standaardiseren om onderlinge uitwisseling van de modulen mogelijk te maken. Toch zijn verschillen onvermijdelijk. Denk maar aan de woordbreedte die 4 bit, 8 bit of 16 bit kan zijn. Wanneer met pariteiten wordt gewerkt worden deze woordbreedten nog eens met één verhoogd.
Texas Instruments heeft op dit gebied veel werk verricht en geldt als toonaangevende leverancier. In figuur 3/6.17-9 is de overeenkomst tussen 4-bit SIP's getekend, die respectievelijk 64 kB x 4, 256 kB x 4 en 1 MB x 4 groot zijn. Op de eerste module zijn 4 DRAM's van 64 kB x 1 gebruikt. Bij de tweede zijn 256 kB x 1 DRAM's toegepast en op de derde bevinden zich 1 MB x 1 DRAM-IC's.

Men ziet dat de aansluitingen van de modulen met elkaar overeen komen (compatibel zijn).

6.17 Toepassen van geheugen-modulen in computers

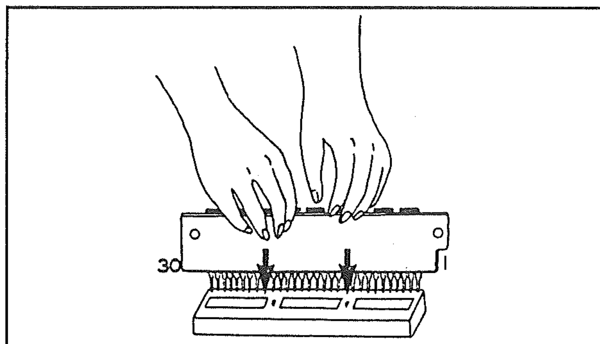


Figuur 3/6.17-10: Overeenkomst tussen 8 bit brede SIP's.

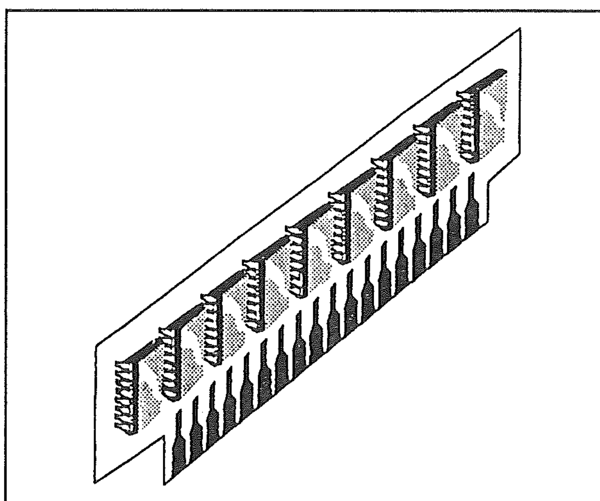
Bij het tweede type wordt pen 1 voor adreslijn A8 gebruikt, terwijl bij het derde de toevoeging van een extra aansluitpen noodzakelijk was. Wanneer nu op de moederprint van de PC de connectoren en de bedrading voor het grootste type worden aangebracht, kunnen de kleinere typen daar alvast gebruik van maken. Bij beschikbaar komen van de grotere typen kunnen die dan zonder meer in dezelfde connectoren worden gestoken.
Door deze benadering kan dezelfde "pinout" ook worden toegepast voor SIP's met grotere woordbreedten, zoals in figuur 3/6.17-10 wordt getoond. Van de V_{DD}-pen tot de V_{SS}-pen zijn alle aansluitingen gelijk aan die van figuur 3/6.17-9. Alleen zijn er aan beide uiteinden nieuwe aansluitingen bijgekomen, aan de V_{SS}-kant voor de data-lijnen en aan de V_{DD}-kant voor de adreslijnen.
Het is dus zelfs zo dat SIP's van het X4-type (gedeeltelijk) van deze connectoren gebruik kunnen maken, als er maar op wordt gelet dat de gelijknamige aansluitingen op de juiste plaats komen.

De SIP's hebben als nadeel dat het vrij moeilijk is de modulen op de basisprint aan te brengen.
Het kan vrij gemakkelijk gebeuren dat een pennetje krom gaat staan en niet in het contact van de socket valt! Daar moet men dus goed op letten!

6.17 Toepassen van geheugen-modulen in computers



Figuur 3/6.17-11: Het monteren van een SIP in de socket van de moederprint van de PC.



Figuur 3/6.17-12: De uiterlijke verschijningsvorm van een SIMM.

Men doet er verstandig aan, zoals getekend in figuur 3/6.17-11, de SIP met beide handen beet te pakken en voorzichtig in de socket te duwen. Daarbij moet men er op letten dat de SIP steeds volledig haaks blijft staan op de moederprint van de PC.

SIMM's

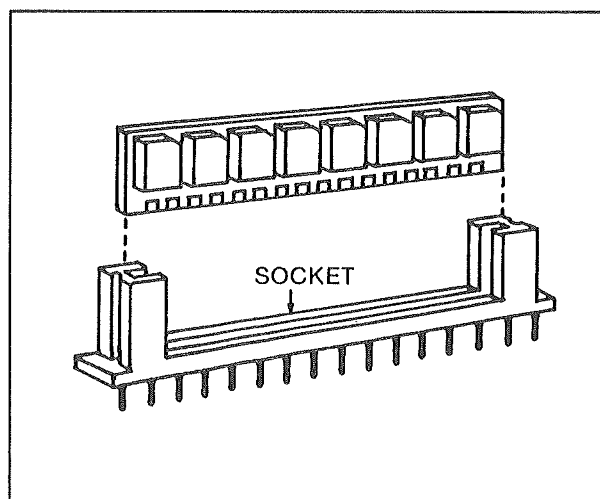
Om de moeilijke, zij het eenmalige, montage van SIP's te vermijden en tegemoet te komen aan de wens van veel PC-gebruikers en -fabrikanten worden nu meestal SIMM's toegepast. Dit zijn in feite dezelfde kleine printjes, maar dan met "edge"-connectoren. Aan de rand van de module zijn vergulde of vernikkelde contactbaantjes geplaatst die in verende contacten van de connector op de moederprint worden gestoken, zie figuur 3/6.17-12. Dit gaat veel gemakkelijker en is bovendien een stuk veiliger.

Het monteren van dergelijke SIMM's gaat volgens figuur 3/6.17-13. Het zal duidelijk zijn dat het monteren van een SIMM zelfs voor de meest onhandige doe-het-zelver een probleemloze klus is!

De moderne standaard

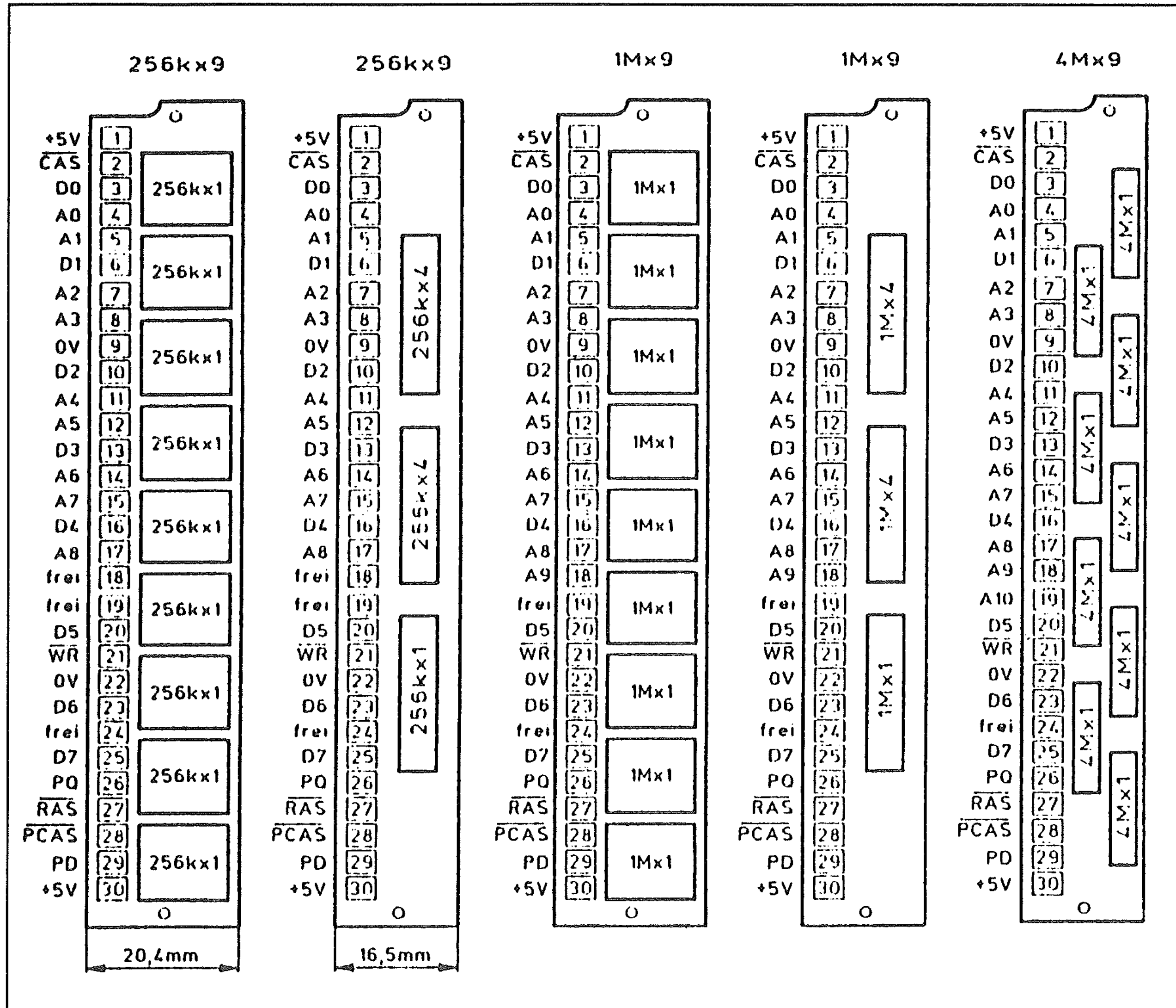
SIMM's als standaard

In alle moderne computer-systemen wordt gebruik gemaakt van 30-polige SIMM's met een geheugenstructuur van 9 x 256 kB, 9 x 1 MB of 9 x 4 MB. Deze modulen hebben gestandaardiseerde aansluitcodes, die in de tabel van figuur 3/6.17-14 worden gepresenteerd.



Figuur 3/6.17-13: Het monteren van een SIMM in de socket op de moederprint van de computer.

6.17 Toepassen van geheugen-modulen in computers



Figuur 3/6.17-14: De gestandaardiseerde aansluitgegevens van de moderne 30-polige SIMM's.

De penbenamingen

In het kort een bespreking van de functie van de penbenamingen die men in de diverse data-boeken kan tegen komen:

- V_{SS}
  Massa
- V_{CC}
  Voedingsspanning
- CAS
  Strobe voor de kolommen van de adressen-matrix
- RAS
  Strobe voor de rijen van de adressen-matrix
- D0 - D7 of DQ1 - DQ7
  Data in- en uitgangen
- A0 - A9
  Adres ingangen
- W
  Schrijf- of leespuls
- PCAS of CAS9
  CAS voor het parity bit
- PD of D9
  Data-ingang voor het parity bit

- PQ of Q9
  Data-uitgang voor het parity bit
- NC
  Niet verbonden

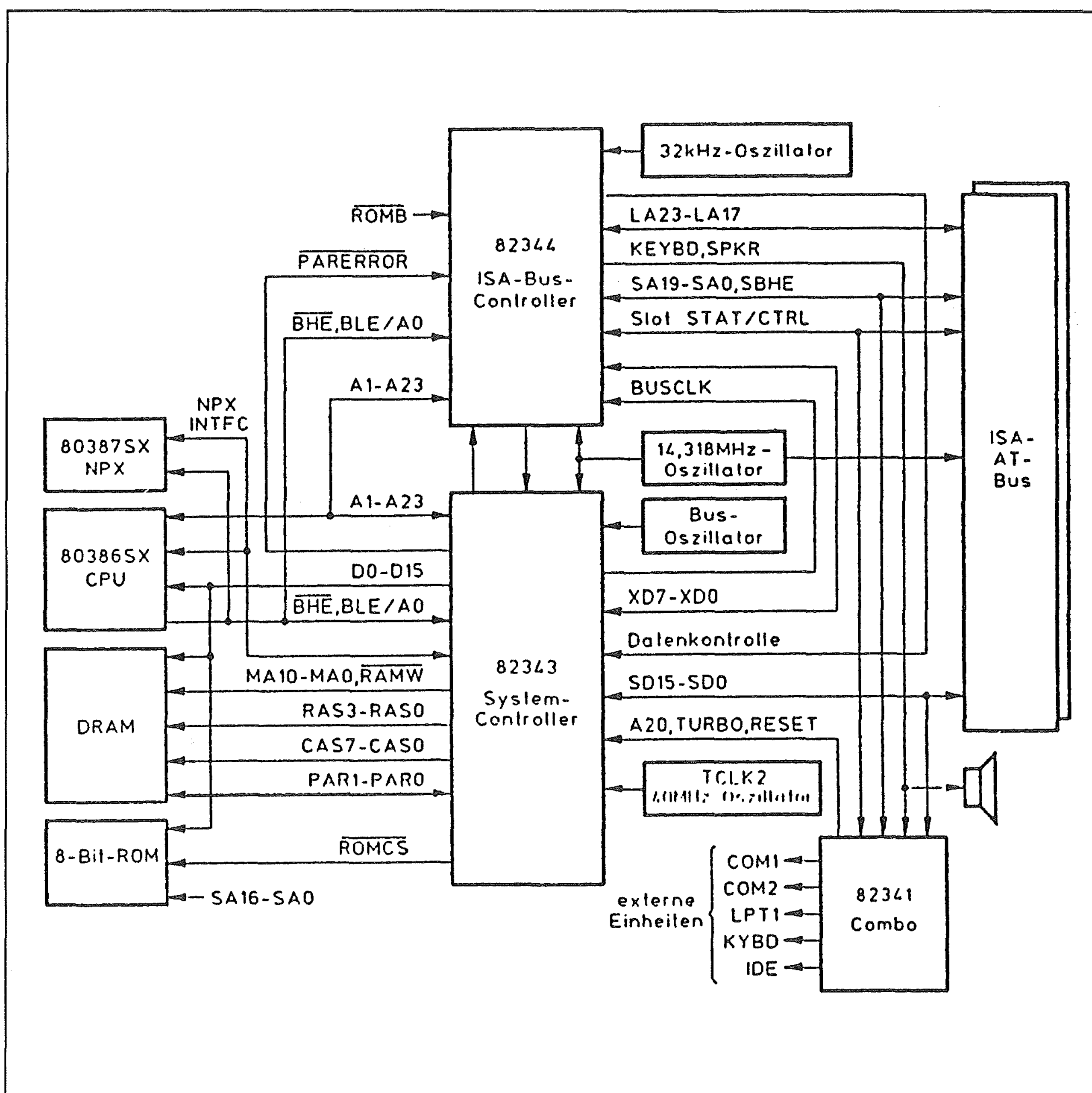
Opmerking

Van sommige 30-pens modulen zijn zowel de SIP- als de SIMM-versie verkrijgbaar.

Deze zijn dan niet alleen elektrisch identiek, maar ook wat betreft de volgorde van de aansluitingen.

De techniek achter het geheugen

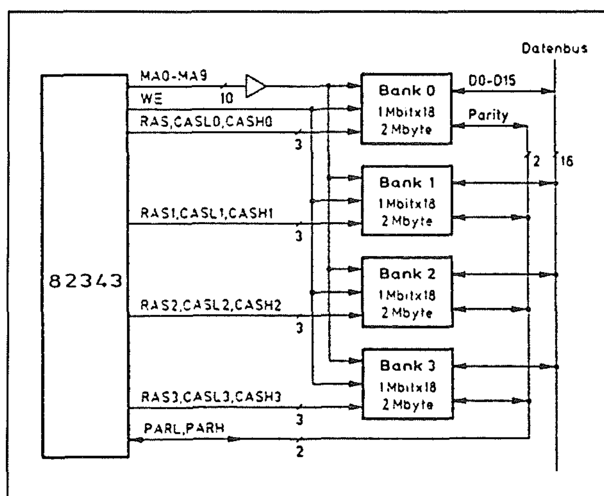
Een moderne PC ziet er blokschematisch uit zoals getekend in figuur 3/6.17-15.



Figuur 3/6.17-15: Het blokschema van een moderne PC met 386-processor.

6.17 Toepassen van geheugen-modulen in computers

De volledige besturing van het geheugen wordt verzorgd door slechts één IC en wel de 82343. Dit IC leest en beschrijft het geheugen, zorgt voor de refresh en voor de parity generatie en controle. Maar de schakeling is zo intelligent dat er, bij vaststellen van meer dan 1 MB aan geheugen bij de opstart-test, onmiddellijk schaduw-RAM wordt vrijgemaakt voor het kopiëren van de BIOS-routines uit het trage ROM-geheugen naar het veel snellere RAM-geheugen.



Figuur 3/6.17-16: Het besturen van de vier geheugen banken uit de 82343.

De vier geheugen banken, die in iedere PC aanwezig zijn, worden volgens figuur 3/6.17-16 door de 82343 gestuurd.
De adressering van het gehele geheugen kan op een van de onderstaande manieren gebeuren:
- Single Bank Page Modus;
- Page Interleave Modus.

In het eerste geval bestaat een geheugenpagina uit 512 adressen als men met 256 kB modulen werkt of uit 2.048 adressen als men met 1 MB modulen werkt.
Bij de Page Interleave Modus worden de gegevens weggeschreven in twee geheugen-banken, waardoor de toegangstijd verkort. Een en ander heeft wél tot gevolg dat men niet op een willekeurige manier de vier banken 0 tot en met 3 met geheugen-modulen mag vullen! Ieder systeem vereist een bepaalde vulvolgorde die in het handboek van de moederprint toegelicht wordt. Ook uit het handboek moet men de stand van enige jumpers op de moederprint voor iedere gewenste geheugen-configuratie afleiden.

Belangrijke opmerking
Het is van het grootste belang dat in alle socket's identieke geheugen-modulen worden geïnstalleerd!
Vaak heeft dit tot gevolg dat men de waarschijnlijk aanwezige 2 MB geheugen zal moeten verwijderen (bijvoorbeeld omdat deze is uitgevoerd onder de vorm van spotgoedkope 256 kB modulen) en dat men nadien alle banken, volgens de volgorde in de handleiding, weer met modulen met een grotere capaciteit zal moeten opvullen.

De snelheid van de geheugen-modulen
Iedere dynamische RAM-chip wordt gekenmerkt door een bepaalde toegangstijd. Deze wordt uitgedrukt in ns, miljoensten van een seconde. Deze tijd zegt iets over de snelheid waarmee nieuwe gegevens in de chip kunnen worden ingelezen.

Hoe sneller de klok van de processor, hoe snellere RAM's er in het geheugen nodig zijn. Er bestaat dus een bepaald verband tussen de kloksnelheid van het systeem en de noodzakelijke minimale toegangstijd van de dynamische RAM's en dus ook van de met deze dynamische RAM's uitgeruste geheugen-modulen.
Dit verband is toegelicht in de tabel van figuur 3/6.17-17.

6.17 Toepassen van geheugen-modulen in computers

KLOKSNELHEID (MHz)	TOEGANGSTIJD GEHEUGEN-IC'S (ns)
4,77	210
8	125
10	100
12	83
16	63
20	50
25	40
36	35
50	25

Figuur 3/6.17-17: Het verband tussen de kloksnelheid van de processor en de minimale toegangstijd van de dynamische RAM's op de geheugen-modulen.

Vaak zal men in de geleverde configuratie tragere modulen aantreffen dan deze die in de tabel vermeld staan.
Snelle dynamische RAM's zijn immers nog steeds zeer duur. Dus heeft men allerlei elektronische truukjes verzonnen om een snelle processor toch met trage RAM's te laten werken.
Enige van deze truukjes zijn:

- Caching
  De reeds beschreven techniek van het opnemen van een klein, snel geheugen tussen processor en systeem-geheugen.
- Introductie van wait-states
  De processor doorloopt een of meerdere wachtlussen tussen schrijf- en leescycli naar het systeem-geheugen.
  In feite komt dit neer op het kunstmatig verlagen van de processorsnelheid bij interacties tussen processor en systeem-geheugen.
- Interleaving
  Door een ingewikkelde structuur tussen processor en systeem-geheugen zal de processor bloksgewijs toegang krijgen tot het geheugen. Hierdoor wordt voorkomen dat snel achter elkaar uit een identieke geheugenlocatie wordt gelezen. Op deze manier staan de data langer op de databus van het blok, zodat ook tragere RAM's in staat zijn om de gegevens op te nemen of weer te geven.

Het gevolg van deze truukjes is dat het wiskundig te berekenen verband tussen kloksnelheid en toegangstijd van de RAM's in de praktijk niet altijd geldt. Nu wil het wel eens gebeuren dat men, uit economisch motieven en ondanks het invoeren van een van de genoemde truukjes, RAM-IC's op de hoofdkaart zet die nét snel genoeg zijn om met de processor te kunnen communiceren. Maar specificaties van IC's zijn niet statisch maar dynamisch. Dat wil zeggen dat bijvoorbeeld de toegangstijd van een IC's iets slechter kan worden als de temperatuur van de chip enige tientallen graden stijgt. En die kleine vergroting van de toegangstijd van slechts één IC op één geheugen-module kan voldoende zijn om regelmatig parity errors te genereren.

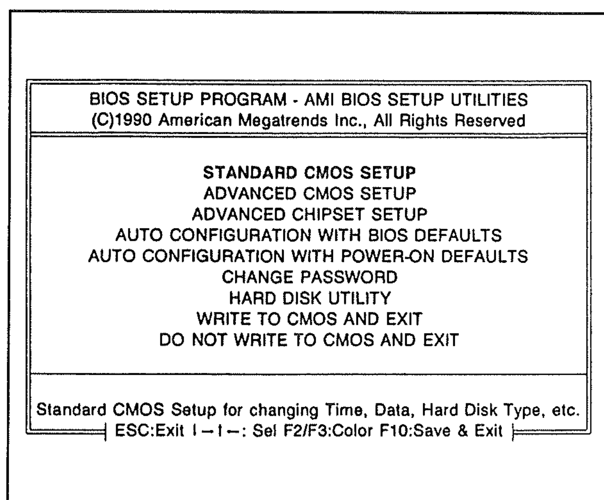
De algemene conclusie moet dan ook luiden dat het nooit kwaad kan te snelle dynamische RAM's in het geheugen op te nemen, maar dat het omgekeerde ten stelligste af te raden is!

SETUP
Na het monteren van de SIMM's op de moederprint zal men in de meeste gevallen de SETUP van het systeem moeten veranderen.
In principe zijn er wel systemen die, bij de zelf-test na het opstarten, de grootte van het geheugen bepalen en in het SETUP-

geheugen schrijven, maar in de meeste gevallen zal dit handmatig moeten gebeuren.

De SETUP bevat fundamentele gegevens over de hardware-samenstelling en -werking van het systeem. Deze gegevens worden opgeslagen in een klein CMOS-geheugentje, dat door middel van een (herlaadbaar) batterijtje steeds onder spanning wordt gehouden.

Hoe een en ander in zijn werk gaat is geheel afhankelijk van de maker van het BIOS van het systeem. Bij AMI (American Megatrend Incorporated), een zeer bekende BIOS-leverancier, wordt de SETUP opgeroepen door tijdens het opstarten van het systeem de DELETE-toets ingedrukt te houden. Nadien verschijnt het openingsscherm van de SETUP, zie figuur 3/6.17-18, op het scherm.



Figuur 3/6.17-18: Het openingsscherm van de SETUP van de AMI BIOS.

De eerste keuze uit het menu, "STANDARD CMOS SETUP", geeft toegang tot het beeld dat is weergegeven in figuur 3/6.17-19.

In dit beeld kan men met de cursortoetsen heen en weer wandelen en met de page-up en page-down toetsen andere informatie invoeren. Nadien verlaat men deze SETUP met de ESC-toets en gaat via het hoofdmenu naar de selectie "WRITE TO CMOS AND EXIT". De gewijzigde gegevens worden opgeslagen in het batterij-gevoede geheugentje en blijven dus bewaard. Nadien start het system opnieuw op met, zo is te hopen, automatisch toegang tot alle nieuwe MB's aan geheugen!

Enige voorbeelden van geheugen-modulen

Inleiding

Tot slot van dit hoofdstuk zullen in het kort de specificaties van enige vaak toegepaste moderne geheugen-modulen van Texas Instruments worden besproken.

TM 4256GP9

De TM4256GP9 is een 2,25 MB dynamisch RAM-module met een 256 kB x 9 bit organisatie. De hiervoor benodigde 9 DRAM's (256 kB x 1, plastic chipcarrier) en 9 ontkoppelcondensatoren zijn gemonteerd op een 30-pens SIMM-module. Om de hoogte van de module te beperken zijn de componenten aan beide zijden van het printje geplaatst. Het negende bit (D9, Q9) wordt meestal voor pariteitscontrole gebruikt en heeft een eigen $\overline{CAS}$-besturing. De SIMM-module is voorzien van een presence detect uitgang op de volgens de aansluit-standaard van figuur 3/6.17-14 vrije pen 24 die via een externe weerstand met V_{CC} moet worden verbonden. Zodra een module aanwezig is, gaat deze uitgang dan LAAG.

6.17 Toepassen van geheugen-modulen in computers

BIOS SETUP PROGRAM - STANDARD CMOS SETUP

(C)1990 American Megatrends Inc., All Rights Reserved

```
Date (mn/date/year): Mon, Jun 14 1991 Base memory : 640 KB
Time (hour/min/sec): 11 : 18 : 20      Ext. memory : 15360 KB
Daylight saving    : Disabled   Cyln Head WPcom LZone Sect Size
Hard disk C: type  : 7           462  8    256   511   17  31 MB
Hard disk D: type  : Not Installed
Floppy drive A:    : 1.2 MB, 5,"
Floppy drive B:    : Not Installed
Primary display    : Monochrome
Keyboard           : Installed
```

Month : Jan, Feb,.....Dec
Date : 01, 02, 03,...31
Year : 1901, 1902,...2099

Sun	Mon	Tue	Wed	Thu	Fri	Sat
24	25	26	27	28	1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31	1	2	3	4	5	6

ESC:Exit ↑–↓–:Select F2/F3:Color PU/PD:Modify

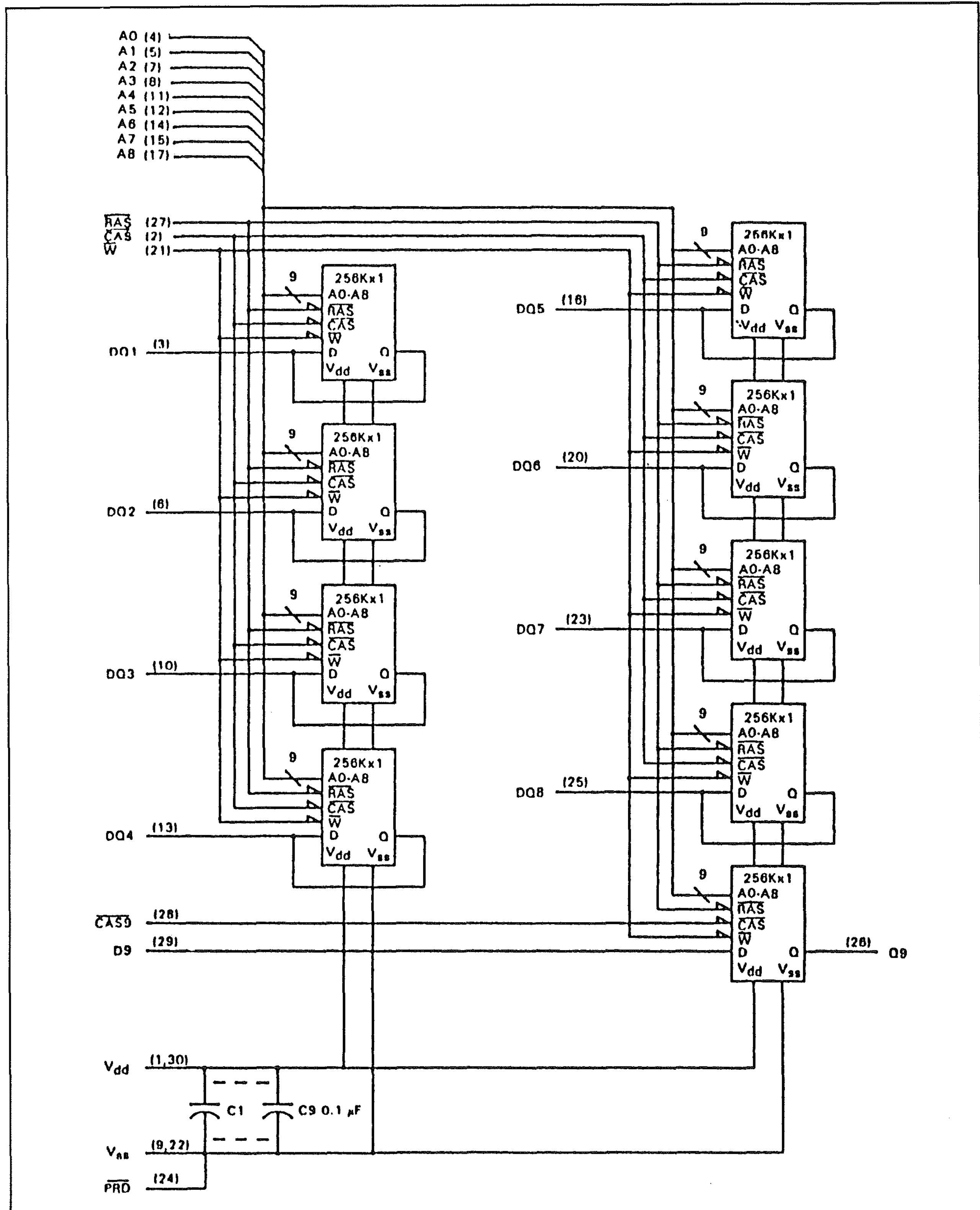
Figuur 3/6.17-19: De "STANDARD CMOS SETUP" geeft informatie over de hoeveelheid geïnstalleerd geheugen.

De specificaties van deze geheugen-module in kort bestek:

- 262.144 x 9 bit organisatie;
- gemeenschappelijke $\overline{CAS}$- en $\overline{RAS}$-ingangen en data in- en uitgangen (3-state) voor 8 bit;
- aparte $\overline{CAS}$9-ingang voor 9e bit;
- gescheiden data in- en uitgang voor 9e bit;
- presence detect ($\overline{PRD}$) uitgang;
- alle in-/uitgangen en clocks TTL-compatibel;
- enkele voeding van +5 V +/-10 %;
- lange refresh-periode van 4 ms max. (256 cycli);
- bevat 9 DRAM's met de onderstaande toegangstijden:
  TMS425x-10: 100 ns
  TMS425x-12: 120 ns
  TMS425x-15: 150 ns
- opwaarts compatibel met 1 MB x 9 modulen en neerwaarts compatibel met 64 kB x 9 modulen.

Het interne schema van dit geheugen-module is getekend in figuur 3/6.17-20.

6.17 Toepassen van geheugen-modulen in computers



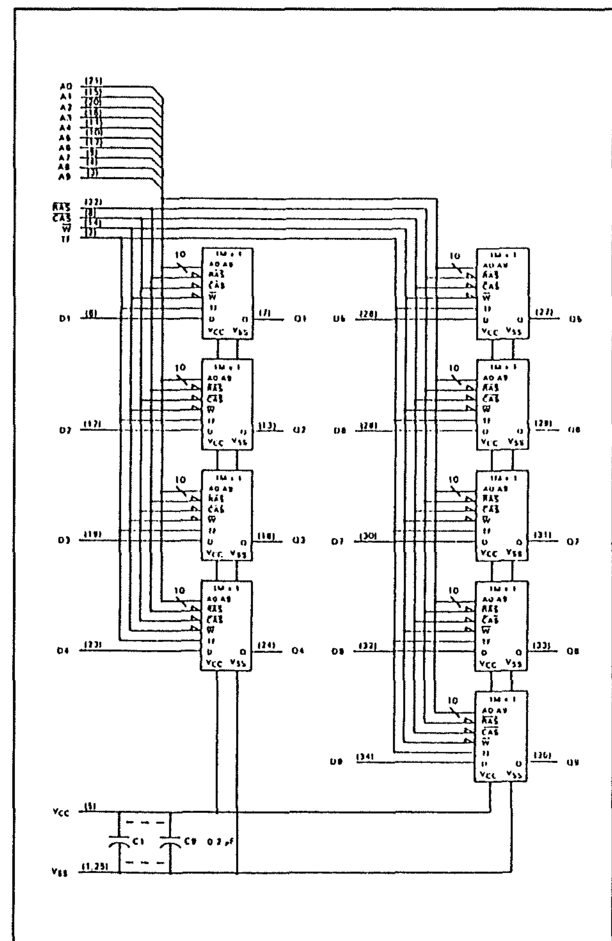
Figuur 3/6.17-20: Functioneel blokschema van de TM4256GP9.

6.17 Toepassen van geheugen-modulen in computers

TM 025EAD9

De TM 025EAD9 is een 9 MB dynamisch RAM-module met een 1.048.576 x 9 bit organisatie. De hiervoor benodigde 9 DRAM's (1 MB x 1, plastic 26-pens SOJ-behuizing) zijn met 9 ontkoppelcondensatoren van 200 nF op een 30-pens SIMM-socket geplaatst. Deze module is leverbaar met naar keuze één van drie soorten DRAM's: TMS4C1024DJ (enhanced page-mode), TMS4C1025DJ (nibble-mode) of TMS4C1027DJ (statische kolom decodeer-mode). De 9e bit heeft een aparte in- en uitgang met bijbehorende $\overline{CAS9}$.

Specificaties in het kort:
- 1.048.576 x 9 bit organisatie
- gemeenschappelijke $\overline{CAS}$- en $\overline{RAS}$-ingangen;
- de 8 data-bits hebben gecombineerde data in- en uitgangen (3-state);
- het 9e bit heeft gescheiden in-/uitgang en $\overline{CAS9}$;
- alle in-/uitgangen en clocks zijn TTL-compatibel;
- enkele voeding nodig van +5 V +/-10 %;
- lange refresh-periode van 8 ms max. (512 cycli);
- bevat 9 DRAM's:
  TM02_EAD9/GAL9-10 (enhanced page-mode): 100 ns
  TM02_EAD9/GAL9-12 (nibble-mode): 120 ns
  TM02_EAD9/GAL9-15 (static column-mode): 150 ns
- geringe dissipatie;
- neerwaarts compatibel met 256 kB x 9 en 64 kB x 9 SIMM's.



Figuur 3/6.17-21: Functioneel blokschema van de TM 025EAD9.

6.17 Toepassen van geheugen-modulen in computers

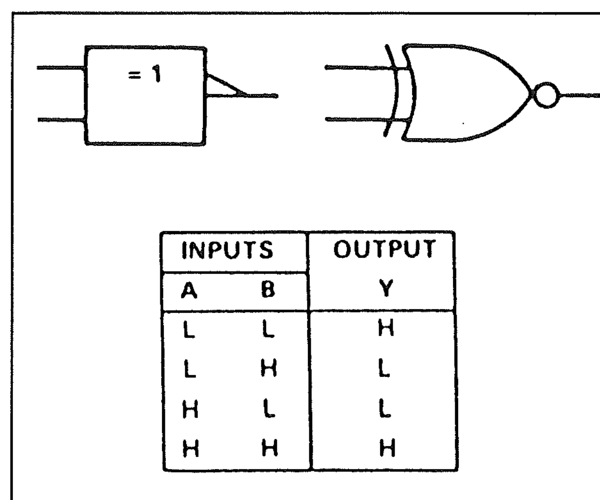
3/6.18
Digitale comparatoren

Inleiding

Codes vergelijken
Comparatoren zijn in het algemeen schakelingen die de waarde van twee signalen met elkaar vergelijken. De comparator geeft een uitgangssignaal af als het ene ingangssignaal groter of kleiner wordt dan het andere ingangssignaal. In de analoge techniek worden comparatoren veelvuldig toegepast. Deze analoge schakelingen hebben digitale soortgenoten, de digitale comparatoren. Hoewel deze schakelingen niet erg bekend zijn, worden zij toch vaak toegepast.

Binair vergelijken
Digitale comparatoren worden gebruikt om twee binaire woorden met elkaar te vergelijken. De eenvoudigste melden alleen wanneer deze woorden gelijk zijn, terwijl er ook typen zijn die "kleiner dan" en "groter dan" aangeven.
Afhankelijk van het gebruik worden ze magnitude (grootte) comparator, adres comparator of identiteits comparator genoemd.

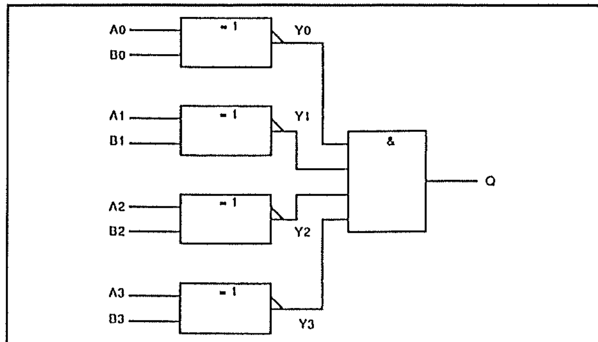
Toepassingsvoorbeeld
In PC's zit een uitbreidingsbus, waarop perifere schakelingen kunnen worden aangesloten. Monteert men bijvoorbeeld de print van een handscanner in de computer, dan zal de processor op de een of de andere manier met de elektronica van de kaart moeten kunnen communiceren. Het is daarvoor noodzakelijk dat aan de kaart een adres wordt toegekend. Dat adres in instelbaar door middel van DIP-schakelaars op de kaart. Wil het systeem met de scanner communiceren, dan wordt het adres van de kaart op de adresbus gezet. Op de kaart is een digitale adrescomparator aanwezig, die de adrescode vergelijkt met de code die met de DIP-schakelaars is ingesteld. Als beide gegevens identiek zijn wordt de elektronica van de kaart geactiveerd.



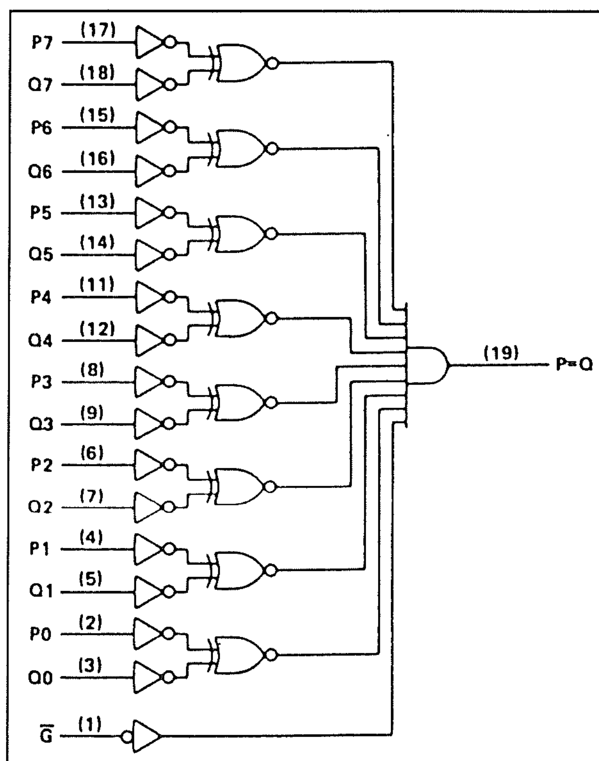
INPUTS		OUTPUT
A	B	Y
L	L	H
L	H	L
H	L	L
H	H	H

Figuur 3/6.18-1: Het logisch symbool en de waarheidstabel van een EXNOR-poort.

6.18 Digitale comparatoren



Figuur 3/6.18-2: Schema voor het vergelijken van twee 4 bit brede woorden.



Figuur 3/6.18-3: Functioneel blokschema (positieve logika) van de 8 bit comparator 74ALS518.

De fundamentele digitale comparator

De eenvoudigste vorm van een binaire comparator is de Exclusive-NOR poort, getekend in figuur 3/6.18-1.

Zoals uit de waarheidstabel blijkt, geeft de EXNOR-poort aan wanneer twee 1 bit brede getallen A en B aan elkaar gelijk zijn (zowel beide “L” als beide “H”). De uitgang van de poort gaat dan naar “H”.
Het is met deze eenvoudige schakeling wel mogelijk te signaleren dat A ongelijk is aan B, maar niet dat A > B (A groter dan B) of A < B (A kleiner dan B) is.

Meervoudig vergelijken

Met een aantal EXNOR-poorten parallel en een AND-poort kunnen grotere binaire getallen (woorden) met elkaar worden vergeleken. Figuur 3/6.18-2 geeft een voorbeeld voor het vergelijken van twee 4 bit brede woorden. Wanneer het A-woord gelijk is aan het B-woord zijn alle vier Y-uitgangen van de EXNOR's “H”, zodat ook de uitgang Q van de AND-poort “H” wordt. In alle andere gevallen is Q echter “L”.

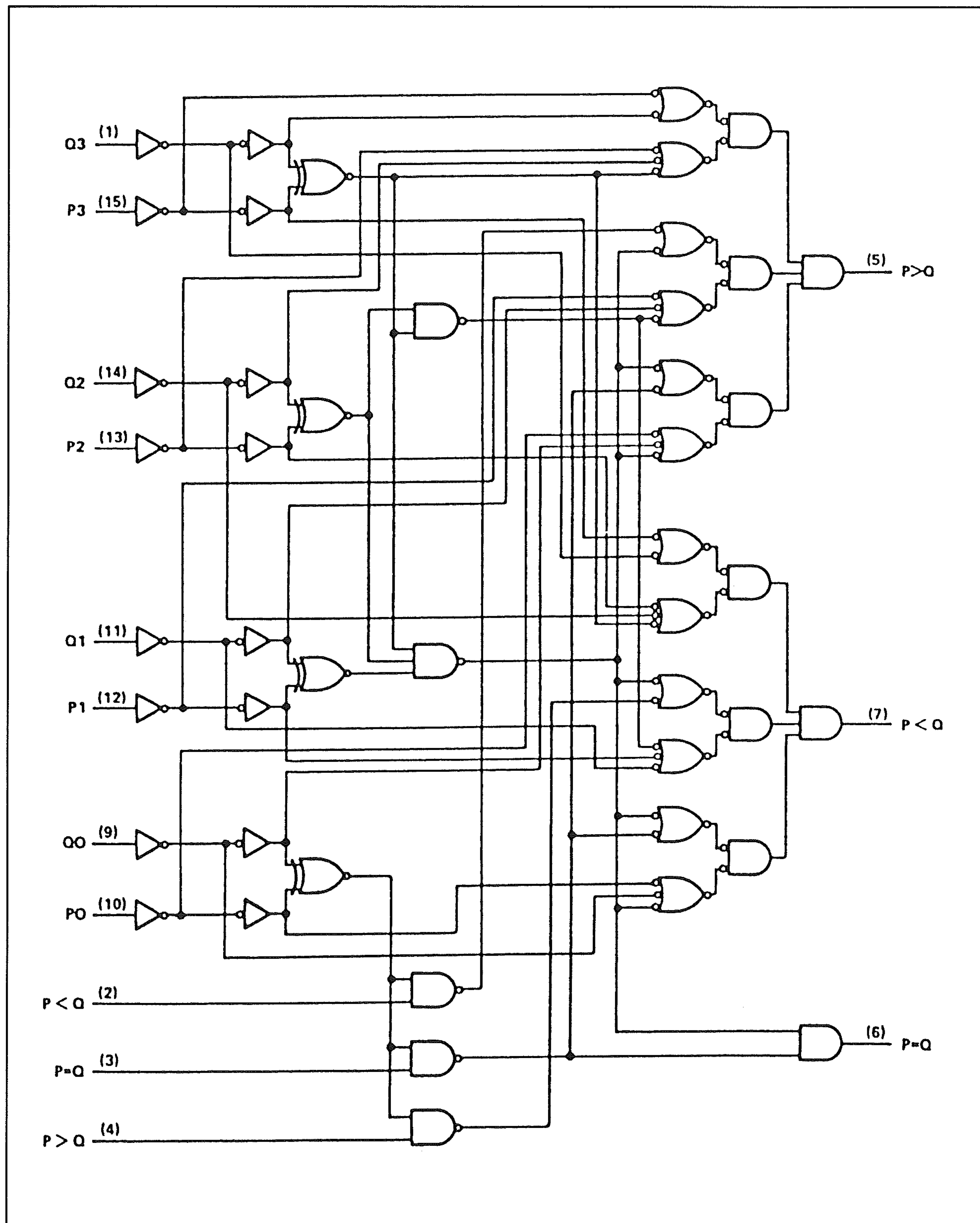
Praktijkvoorbeeld: de 74ALS518

Dat dergelijke schakelingen ook werkelijk in de praktijk worden toegepast bewijst figuur 3/6.18-3, waar het functionele blokschema van de 8 bit brede zogenoemde identiteits-comparator 74ALS518 wordt voorgesteld.
Bij bestudering van het logische schema blijkt dat de uitgang P = Q alleen “H” wordt als het P-woord gelijk is aan het Q-woord.

A gelijk aan, groter dan of kleiner dan B

Dat voor het detecteren van A > B, A = B en A < B een veel ingewikkelder schakeling nodig is, zal duidelijk zijn. In de figuren 3/6.18-4 en 3/6.18-5 worden het functionele blokschema en de waarheidstabel van de 4 bit brede comparator 7485 getoond. Deze schakeling is zelfs nog iets ingewikkelder door de cascade-ingangen waarmee de toestand van naburige comparatoren kan worden doorgegeven.

6.18 Digitale comparatoren



Figuur 3/6.18-4: Het blokschema van de vier bit brede comparator 7485.

6.18 Digitale comparatoren

COMPARING INPUTS				CASCADING INPUTS			OUTPUTS		
P3, Q3	P2, Q2	P1, Q1	P0, Q0	P > Q	P < Q	P = Q	P > Q	P < Q	P = Q
P3 > Q3	X	X	X	X	X	X	H	L	L
P3 < Q3	X	X	X	X	X	X	L	H	L
P3 = Q3	P2 > Q2	X	X	X	X	X	H	L	L
P3 = Q3	P2 < Q2	X	X	X	X	X	L	H	L
P3 = Q3	P2 = Q2	P1 > Q1	X	X	X	X	H	L	L
P3 = Q3	P2 = Q2	P1 < Q1	X	X	X	X	L	H	L
P3 = Q3	P2 = Q2	P1 = Q1	P0 > Q0	X	X	X	H	L	L
P3 = Q3	P2 = Q2	P1 = Q1	P0 < Q0	X	X	X	L	H	L
P3 = Q3	P2 = Q2	P1 = Q1	P0 = Q0	H	L	L	H	L	L
P3 = Q3	P2 = Q2	P1 = Q1	P0 = Q0	L	H	L	L	H	L
P3 = Q3	P2 = Q2	P1 = Q1	P0 = Q0	X	X	H	L	L	H
P3 = Q3	P2 = Q2	P1 = Q1	P0 = Q0	H	H	L	L	L	L
P3 = Q3	P2 = Q2	P1 = Q1	P0 = Q0	L	L	L	H	H	L

Figuur 3/6.18-5: De waarheidstabel van de vier bit brede comparator 7485.

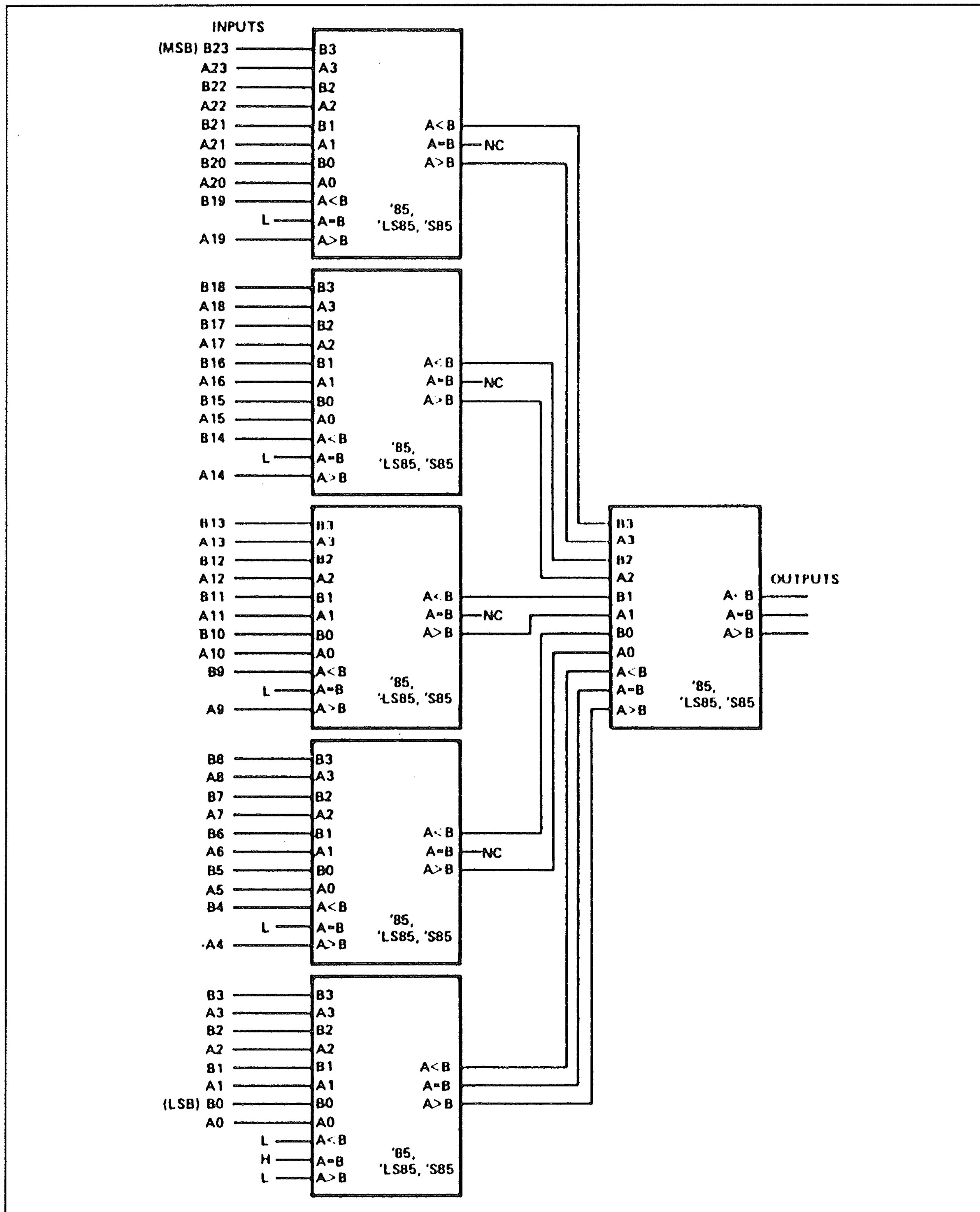
De schakeling heeft drie uitgangen:
- P>Q;
- P=Q;
- P<Q.

Dank zij de cascade-ingangen kan men met identieke schakelingen cascaderen voor het vergelijken van 8, 12 of 16 bit brede woorden. De drie uitgangen van de schakeling die de vier laagste bits vergelijkt worden verbonden met de gelijknamige cascade-ingangen van het daaropvolgende IC.
De A=B ingang van de schakeling die de meest belangrijkste bits vergelijkt moet met de voeding verbonden worden, de twee overige cascade-ingangen gaan naar de massa.

Voorbeeld
Als voorbeeld van de eenvoudige manier waarop met dergelijke schakelingen grote digitale comparatoren samengesteld kunnen worden is in figuur 3/6.18-6 het schema getekend van een digitale comparator, die twee 24 bit brede woorden met elkaar vergelijkt. Dat kan met slechts zes identieke schakelingen van het type 7485!

De uitgangen van de vijf comparatoren, die de ingangsbits vergelijken, gaan naar de ingangen van de zesde comparator en worden daar weer met elkaar vergeleken. Let op de manier waarop de cascade-ingangen van de vijf ingangsschakelingen met de “L” en “H” worden verbonden!

6.18 Digitale comparatoren



Figuur 3/6.18-6: Zes IC's van het type 7485 vormen een 24 bit digitale comparator.

6.18 Digitale comparatoren

Toepassingen

Inleiding

Zoals reeds geschreven in de inleiding worden digitale comparatoren gebruikt voor het vergelijken van binaire getallen. Bekende toepassingen daarvan zijn:
- het detecteren van een adres op een interfacekaart in een computer;
- het detecteren van een codewoord;
- het instellen van een teller.

Het instellen van een kaart-adres

Wanneer de mogelijkheden van een computer worden vergroot door gebruik te maken van interfacekaarten (bijvoorbeeld een PIA-kaart, een floppy-disk controller of een geheugenkaart) moeten deze een eigen uniek adres hebben, zodat de betreffende kaart door het computerprogramma geaktiveerd kan worden. Het instellen van zo'n adres gebeurt meestal met behulp van kleine schakelaars.
In figuur 3/6.18-7 is getekend hoe van een 16 bit breed adres 12 bits door de gebruiker kunnen worden ingesteld. De laagste 4 adresbits worden in dit geval door registers in de IC's op de kaart gebruikt en controleren en besturen de werking van de kaart. Men kan dus 16 functies op de kaart activeren door code-combinaties op deze vier adreslijnen te zetten. De overige 12 bits vormen het kaartadres.
Voor de duidelijkheid worden de te vergelijken woorden aan beide zijden van de comparatoren van het reeds besproken type 7485 ingevoerd, links de adresbits van de bus en rechts de programmaschakelaars.

De lijnen die naar de schakelaars gaan worden door 16 weerstanden met de voeding verbonden en staan dus bij geopende schakelaar op "H". Sluit men een schakelaar, dan wordt de betreffende lijn door de schakelaar met de massa verbonden en gaat dus naar "L".

Alternatief systeem

In plaats van de adresdetector 7485 kan ook een ander type zoals de 74ALS677 worden gebruikt. Het functionele blokschema van dit IC is getekend in figuur 3/6.18-8.

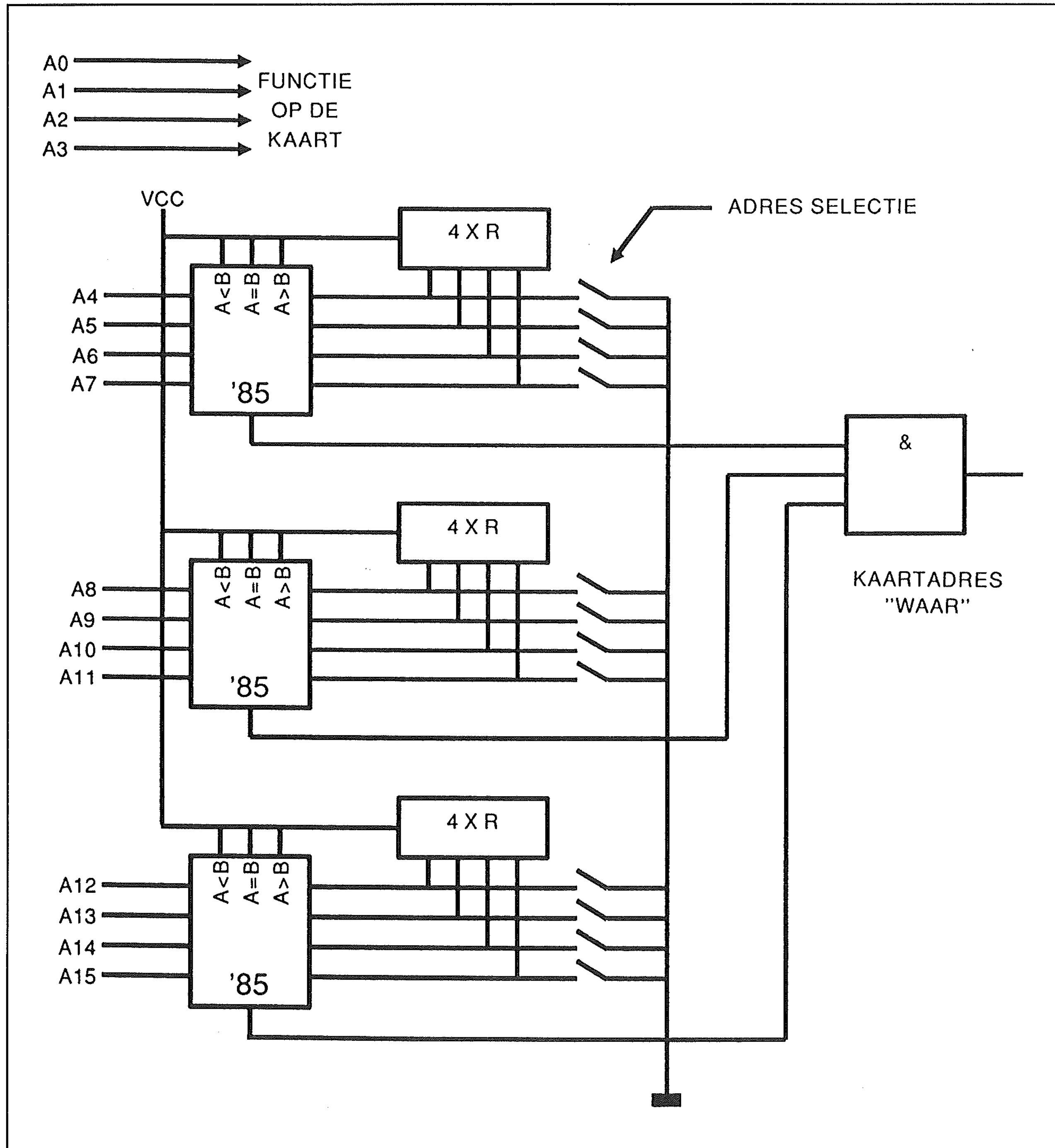
Hierbij worden de adreslijnen van de bus op de ingangen A1 tot en met A16 aangesloten. Met de P0- tot en met P4-ingangen wordt (meestal door een "harde bedrading") het aantal te detecteren "nullen" ingesteld dat door de laagste adresingangen moet worden gedetecteerd. Wordt bijvoorbeeld de combinatie "L-H-H-H" (decimaal 7) op de P-ingangen gezet, dan zijn de laagste 7 adresbits (A1 tot en met A7) ingericht om logische nullen te ontvangen.
De resterende adresbits zijn dan ingesteld om "H"-en te detecteren (zie ook de waarheidstabel 3/6.18-9). Het spreekt vanzelf dat het exacte adres wordt bepaald door de manier van aansluiten van de adreslijnen.

Een praktische schakeling

Tot slot wordt een praktische schakeling van een adres-decoder beschreven, die gebruikt kan worden bij zelfbouw systemen op basis van een 6502-processor (C-64, Apple, etc).
Het volledige schema van deze universele adres-decoder is getekend in figuur 3/6.18-10.

De adreslijnen A2 tot en met A7 van de processor adresbus worden gebruikt voor het instellen van het I/O-adres.

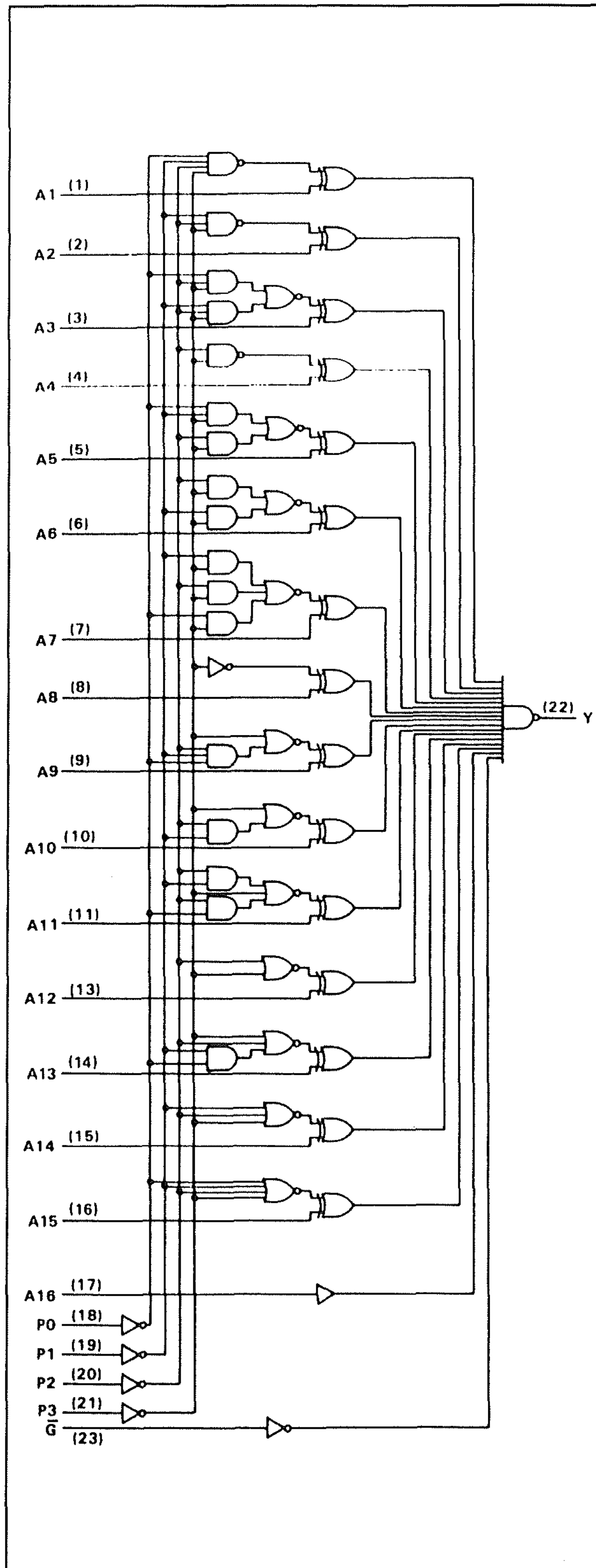
6.18 Digitale comparatoren



Figuur 3/6.18-7: Programmering van 12 van de 16 adresbits op een interfacekaart.

Deze bits worden in twee adresdecoders van het type 74LS85 vergeleken met de standen van acht DIL-schakelaartjes. De A=B uitgang van de onderste comparator gaat via een inverter naar de enable-ingang van een bi-directionele buffer van het type 74LS245. Deze buffer is aangesloten op de data-bus van de processor (links) en op de interne databus van het externe systeem (rechts).

6.18 Digitale comparatoren



Figuur 3/6.18-8: Het blokschema van de 74ALS677.

De richting waarin de bidirectionele buffer werkt wordt bepaald door het R/W signaal van de processor.
Het A=B signaal van de digitale comparator wordt samen met het PHI2 signaal van de processor gebruikt voor het triggeren van een monostabiele multivibrator. Deze stuurt twee LED's aan, waarvan een op de printplaat zit en een eventueel op de frontplaat van de uitbreiding gemonteerd kan worden. Op deze manier heeft men een visuele controle van het aanspreken van de kaart.

Fuse-programmable comparatoren

Inleiding

De besproken schakelingen hebben twee set's ingangen, waarop men signalen moet zetten. Er bestaan echter ook digitale comparatoren, waar een van deze set's signalen in het IC wordt ingebrand. Deze digitale comparatoren noemt men "fuse-programmable" digitale comparatoren en worden gebruikt in toepassingen, waarbij een uniek onveranderlijk adres moet worden gedetecteerd. Dat kan bijvoorbeeld het adres zijn van een magnetische codekaart, waarmee men een slot kan openen.

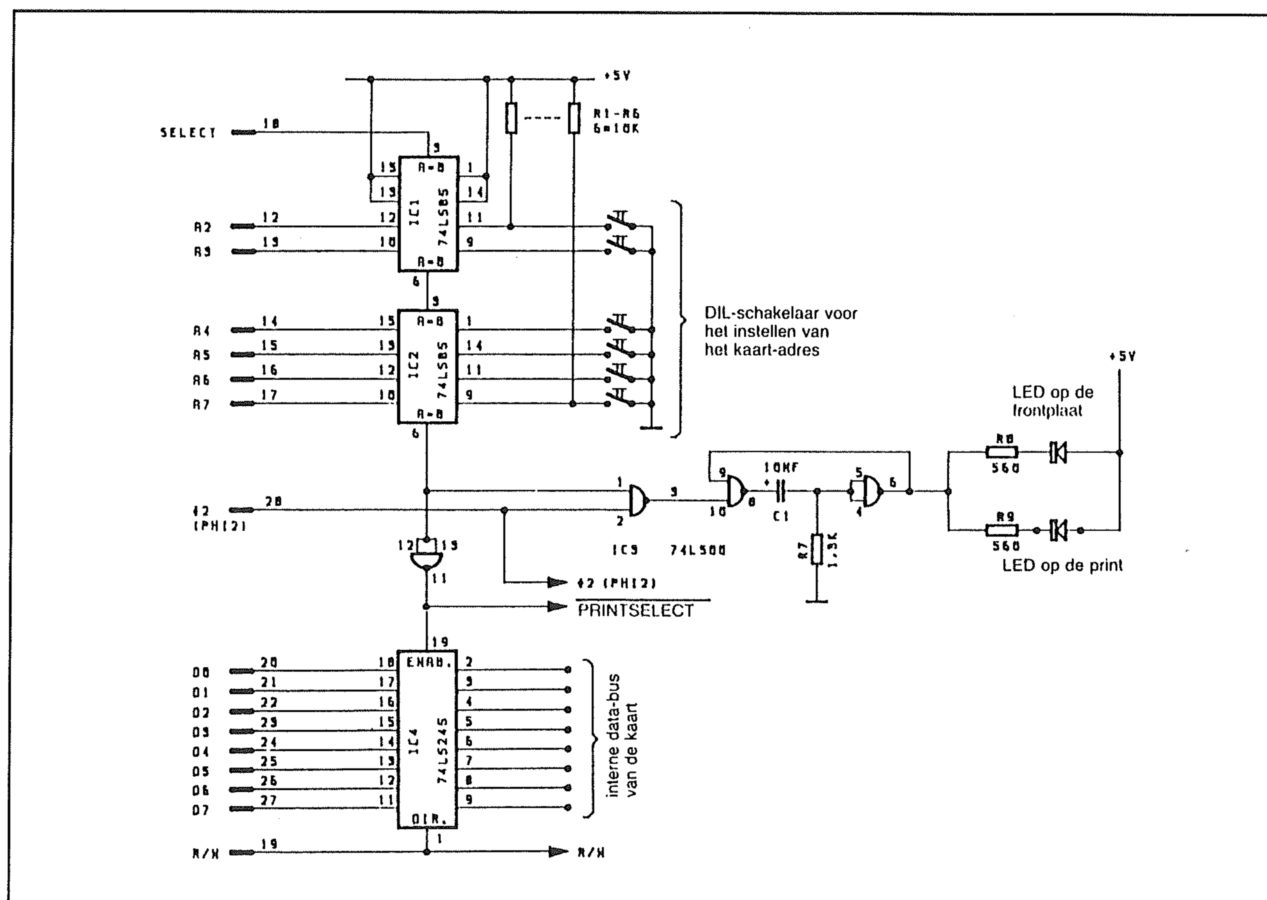
Een voorbeeld: de 74ALS526

Het bekendste voorbeeld van dit soort digitale comparatoren is de 74ALS526, waarvan het blokschema in figuur 3/6.18-11 is getekend.
Bij deze schakeling wordt een 16 bit breed woord (bijvoorbeeld een adres) vergeleken met een van te voren hierin geprogrammeerd datawoord.

6.18 Digitale comparatoren

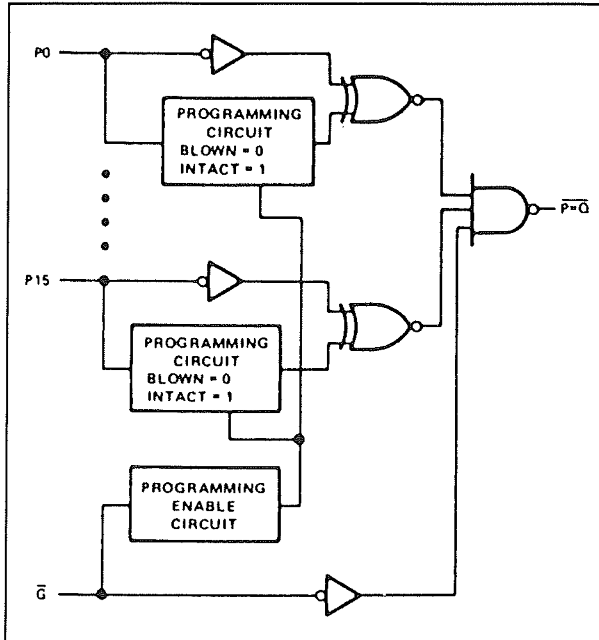
'ALS677	'ALS678	INPUTS COMMON TO 'ALS677 AND ALS678																				OUTPUT
$\overline{G}$	C	P3	P2	P1	P0	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	A14	A15	A16	Y
L	H	L	L	L	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	L
L	H	L	L	L	H	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	L
L	H	L	L	H	L	L	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	L
L	H	L	L	H	H	L	L	L	H	H	H	H	H	H	H	H	H	H	H	H	H	L
L	H	L	H	L	L	L	L	L	L	H	H	H	H	H	H	H	H	H	H	H	H	L
L	H	L	H	L	H	L	L	L	L	L	H	H	H	H	H	H	H	H	H	H	H	L
L	H	L	H	H	L	L	L	L	L	L	L	H	H	H	H	H	H	H	H	H	H	L
L	H	L	H	H	H	L	L	L	L	L	L	L	H	H	H	H	H	H	H	H	H	L
L	H	H	L	L	L	L	L	L	L	L	L	L	L	H	H	H	H	H	H	H	H	L
L	H	H	L	L	H	L	L	L	L	L	L	L	L	L	H	H	H	H	H	H	H	L
L	H	H	L	H	L	L	L	L	L	L	L	L	L	L	L	H	H	H	H	H	H	L
L	H	H	L	H	H	L	L	L	L	L	L	L	L	L	L	L	H	H	H	H	H	L
L	H	H	H	L	L	L	L	L	L	L	L	L	L	L	L	L	L	H	H	H	H	L
L	H	H	H	L	H	L	L	L	L	L	L	L	L	L	L	L	L	L	H	H	H	L
L	H	H	H	H	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	H	H	L
L	H	H	H	H	H	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	H	L
L	H	All other combinations																				H
H																						H

Figuur 3/6.18-9: Waarheidstabel van de 74ALS677.



Figuur 3/6.18-10: Het schema van een universeel bruikbare adres-decoder voor 6502-systemen.

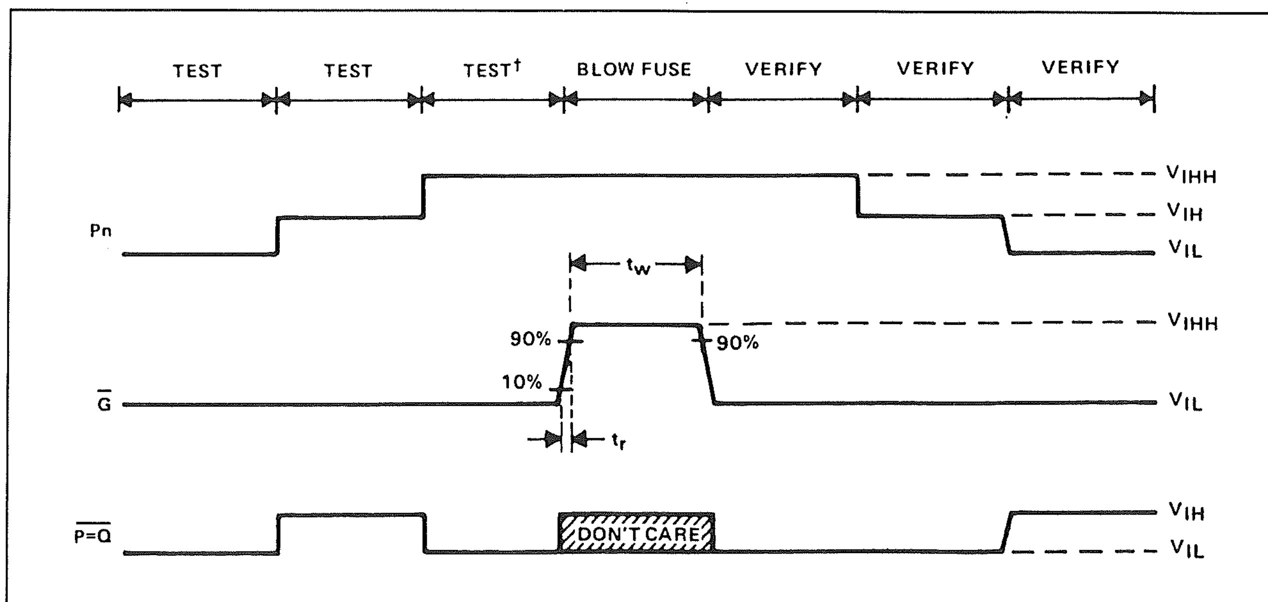
6.18 Digitale comparatoren



Figuur 3/6.18-11: Het logische blokschema van de 74ALS526.

Het programmeren berust op het doorbranden van "zekeringen". De uitgang gaat naar "L" als beide digitale codes identiek zijn.
De bits worden geprogrammeerd (zekeringen doorgebrand) door een spanning van 12 V op zowel de betreffende P-ingang als de G-ingang te zetten. Hiervoor is geen speciale programmeerapparatuur nodig. Let op dat slechts één bit tegelijk mag worden geprogrammeerd en dat het programmeren definitief is. De geprogrammeerde bits kunnen HOGE niveaus detecteren (logisch 1), de niet-geprogrammeerde herkennen LAGE niveaus (logisch 0).

Toepassingen
Op dezelfde wijze dat een adres kan worden gedetecteerd is het natuurlijk ook mogelijk om een willekeurig ander woord voor herkenning in te stellen. Zo kan bijvoorbeeld een elektronisch slot met een cijfercode worden geprogrammeerd.
Een andere toepassing is het tellen tot een bepaalde, in te stellen waarde. Het meest gebruikelijk is het laden van een teller met deze waarde, waarna wordt terug geteld tot nul.
Wanneer dit echter niet mogelijk is, moet worden opgeteld totdat de geprogrammeerde waarde wordt bereikt.



Figuur 3/6.18-12: Het programmeren van een bit bij een fuse-programmable digitale comparator.

3/6.19

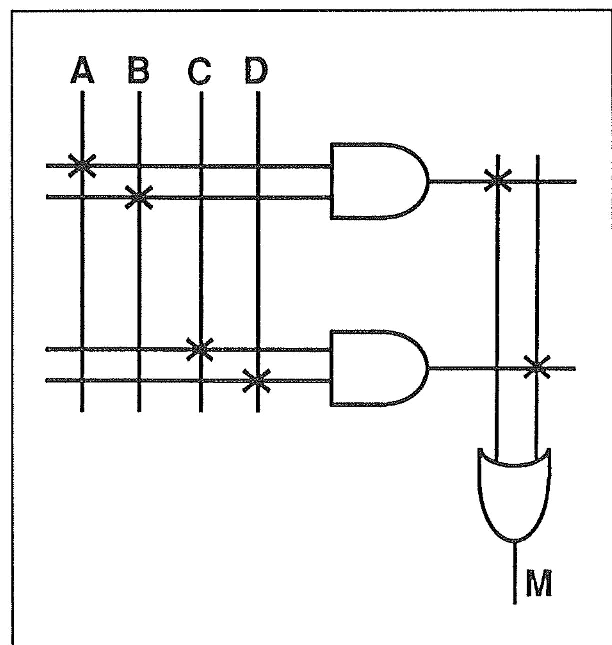
Werking en principes van PLD's

Inleiding

Wat is een PLD?
Het letterwoord PLD is de afkorting van "Programmable Logic Device", programmeerbaar logisch onderdeel. Een PLD is een digitaal geïntegreerde schakeling, die is samengesteld uit vele identieke basiselementen zoals poorten en flip-flop's. Het grote verschil met een gewoon logisch IC is dat de verbindingen tussen deze elementen niet tot stand komen gedurende de fabricage van het IC, maar nadien door de gebruiker. Op deze manier is het mogelijk ingewikkelde logische functies in het IC te programmeren, waardoor heel wat standaard IC's uitgespaard kunnen worden.
PLD's lijken dus een beetje op PROM's, geheugens die eenmalig te programmeren zijn en nadien alleen gelezen kunnen worden. PLD's zijn echter veel universeler, omdat niet alleen code's maar ook logische Booleaanse formules in het IC ingebrand kunnen worden.

Een voorbeeld ter kennismaking
In figuur 3/6.19-1 is een eenvoudig, niet realistisch voorbeeld van de structuur van een PLD getekend. Een dergelijke eenvoudige schakeling wordt uiteraard niet op de markt gebracht, maar het geeft een goede indruk van hoe een PLD er uit ziet.

De schakeling heeft vier ingangen A, B, C en D. Deze vier ingangen kunnen via de matrix van horizontale en vertikale lijnen met de ingangen van twee AND-poorten verbonden worden.



Figuur 3/6.19-1: Een eenvoudig voorbeeld van een PLD.

De twee uitgangen van deze poorten kunnen, via een soortgelijke lijnenmatrix, met de ingangen van een OR-poort verbonden worden. De uitgang van de OR-poort is de uitgang M van de schakeling. De kruisjes op de verbindingen in de matrixen geven aan dat de horizontale lijn doorverbonden is met de vertikale lijn.

Het programmeren gaat ongeveer op dezelfde manier als bij een PROM, door het wegbranden van "zekeringen", die op alle snijpunten van de lijnenmatrixen aanwezig zijn. Op deze manier kunnen op een heel eenvoudige manier logische combinatorische functies in het IC geprogrammeerd worden. Een combinatorische functie is een Booleaanse formule, die beschijft welk logisch niveau een uitgangsgrootheid (bijvoorbeeld M) aanneemt voor een bepaalde combinatie van ingangsniveaus op A, B, C en D.

Voorgeschiedenis

PLD's zijn geen nieuwe ontwikkeling, de eerste schakelingen werden reeds in 1975 door Signetics op de markt gebracht. Toen waren deze primitieve PLD's echter zeer duur en alleen voor schakelingen te gebruiken waar geld geen rol speelde. Door de steeds goedkoper wordende productie en de infiltratie van logische digitale technieken in alle mogelijke gebieden van de elektronica, beleven deze schakelingen echter de laatste vijf jaar een zeer grote bloei. Op dit moment worden minstens 3.000 verschillende PLD's aangeboden door alle belangrijke IC-fabrikanten van de wereld.

De eenvoudigste schakelingen kosten niet veel meer dan standaard IC's uit de TTL-serie, zodat de doe-het-zelver die niet terug schrikt voor het experimenteren met nieuwe technieken PLD's niet vanwege de onbetaalbaarheid links moet laten liggen!

Onoverzichtelijk aanbod

Helaas heeft de commercie erg toegeslagen, waardoor er een onoverzichtelijk aanbod is ontstaan van schakelingen, die allemaal onder de verzamelnaam "PLD" te vangen zijn, maar toch op soms belangrijke punten van elkaar verschillen. Maar daarentegen is het ook zo dat identieke schakelingen onder de meest uiteenlopende namen worden aangeboden.

Een en ander heeft te maken met het feit dat diverse fabrikanten letterwoorden, die in feite functie-omschrijvingen zijn, hebben laten registreren. Andere fabrikanten kunnen deze letterwoorden dan uiteraard niet meer gebruiken, zodat nieuwe letterwoorden worden verzonnen om in wezen dezelfde functies te omschrijven.

Op deze manier zijn letterwoorden als:

- PAL;
- GAL;
- IFL;
- FPLA;
- FPLS;
- FPGA;
- PLA;
- LCA;

ontstaan. Omschrijvingen van PLD's, die vaak maar in details van elkaar verschillen.

In dit hoofdstuk zal een poging worden gedaan deze warboel van begrippen voor de leek te ontrafelen en de belangrijkste hoofdzaken van het begrip "PLD" op een rijtje te zetten.

Een toepassingsvoorbeeld

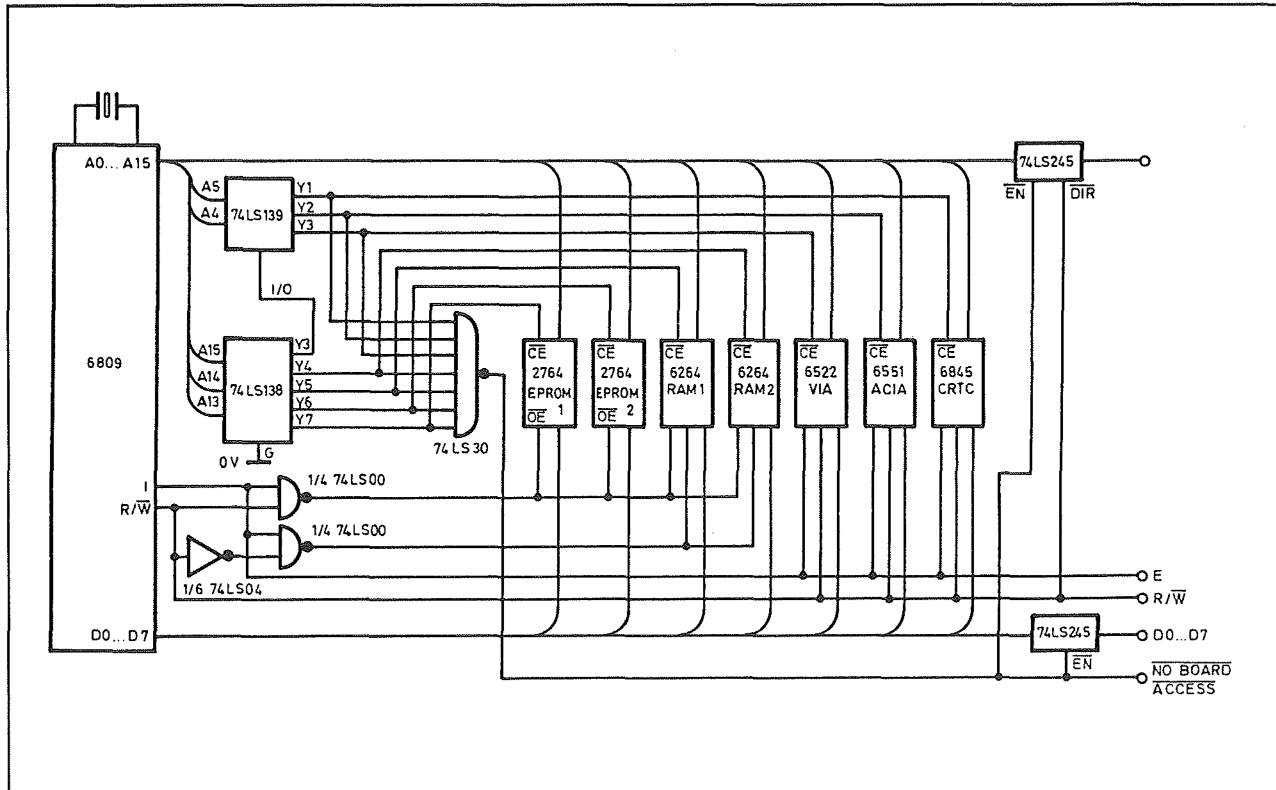
Het praktische belang van PLD's kan het best toegelicht worden aan de hand van een voorbeeldje.

In figuur 3/6.19-2 is een eenvoudig schemaatje getekend van een microprocessorsysteem rond een 6809. Dergelijke kleine microprocessor-systemen worden tegenwoordig in van alles en nog wat toegepast, van magnetrons tot volledig geautomatiseerde temperatuurregelingen.

Een dergelijk systeem bestaat, naast de microprocessor, uit:

- EPROM's waarin het bedrijfssysteem zit;

6.19 Werking en principes van PLD's



Figuur 3/6.19-2: Een klein microprocessor-systeem, waarbij de besturing met standaard IC's uit de TTL-serie wordt uitgevoerd.

- RAM's waarin gebruikersinstructies worden opgeslagen;
- VIA en ACIA waarmee het systeem communiceert met de buitenwereld;
- een CRTC, waarmee een dot-display wordt aangestuurd.

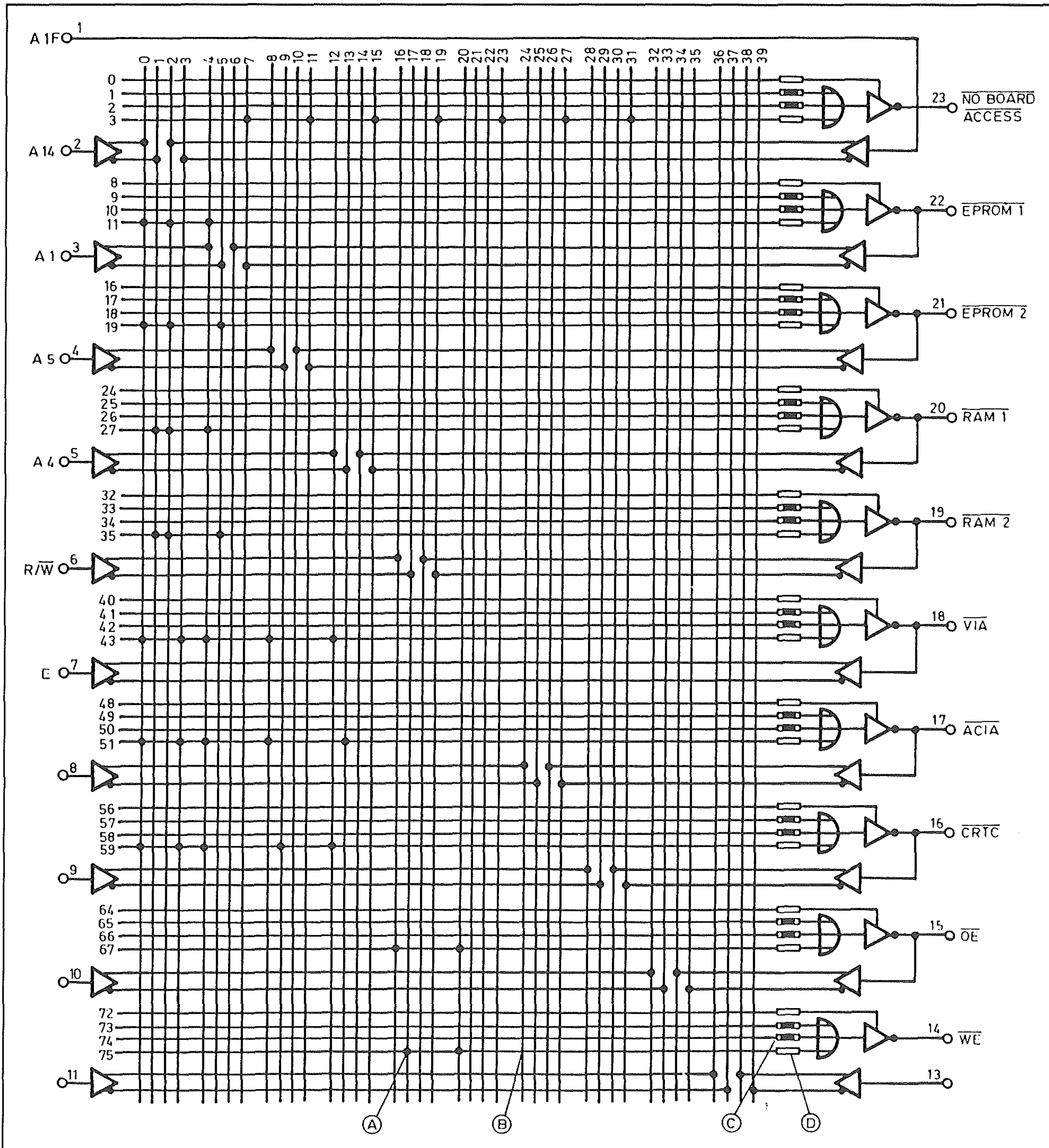
Al die blokken worden geadresseerd door bepaalde digitale codes op de adreslijnen te zetten. Voor het decoderen van deze codes zijn logische schakelingen noodzakelijk, die in het voorbeeld worden uitgevoerd met standaard schakelingen uit de TTL-reeks. In totaal zijn zes poorten noodzakelijk, ondergebracht in vijf IC's. Dit is uiteraard een zeer oneconomische oplossing! Zo worden er bijvoorbeeld van de vier poorten die in de 75LS00 zitten maar twee gebruikt.

In dergelijke systemen, waar miniaturisatie en goedkope productie belangrijk zijn, kan men veel beter gebruik maken van een PLD. De noodzakelijke poortfuncties kunnen worden opgebouwd door de lijnenmatrixen tussen de in- en uitgangen van de basis-elementen in de PLD op de juiste manier met elkaar door te verbinden.

Uit de bestudering van het schema blijkt dat de decodeerlogica zeven ingangen heeft en tien uitgangen. Bovendien kunnen alle functies met AND-schakelingen samengesteld worden. Dit is een klusje waar zelfs de eenvoudigste PLD voor geschapen is!

Het resultaat: in figuur 3/6.19-3 is getekend hoe een PLD van het type 20L10 intern is geprogrammeerd om vanuit de zeven gebruikte ingangen alle tien noodzakelijke besturingssignalen voor het microprocessor-systeem te genereren.

8.9 Software voor de ontwerper



Figuur 3/6.19-3: De volledige besturing van het microprocessor-systeem wordt overgenomen door één PLD van het type 20L10.

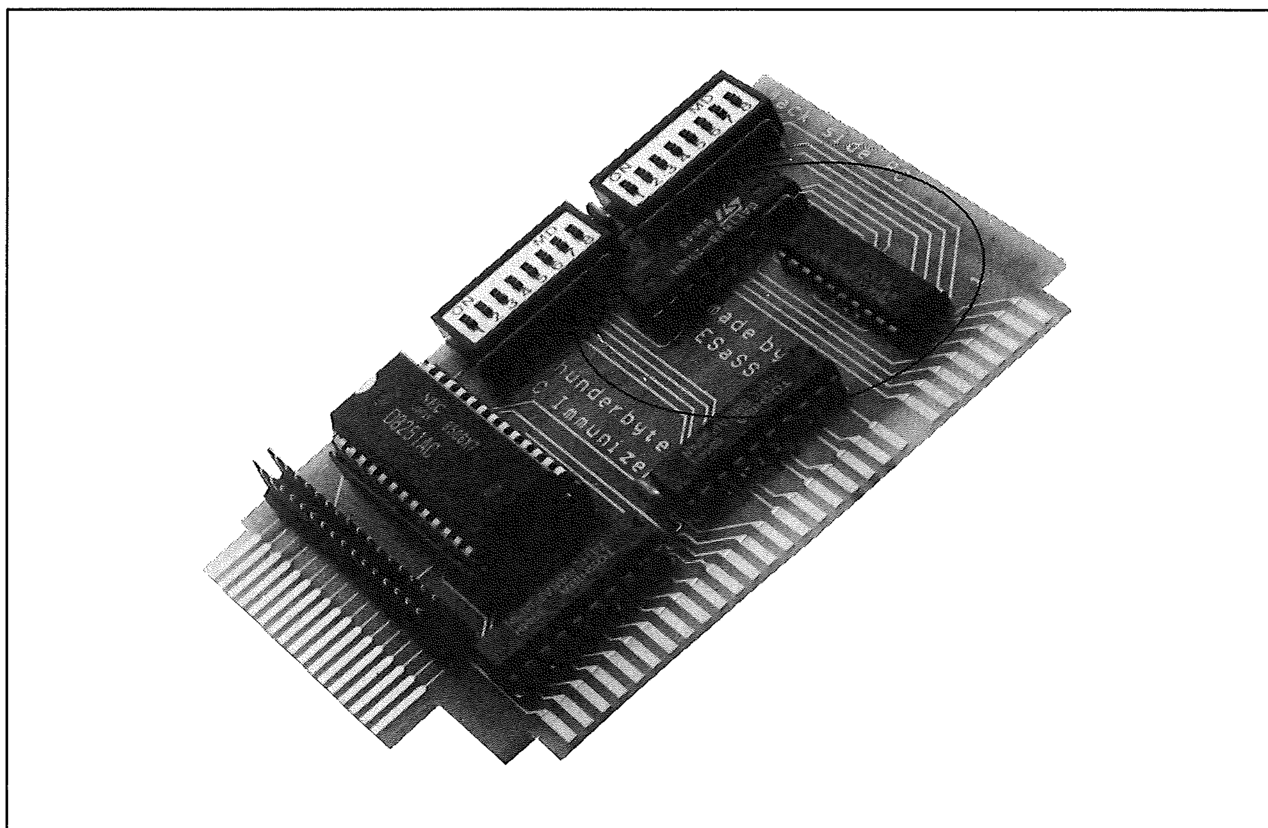
Uit deze figuur blijkt dat de interne opbouw van een praktische PLD niet wezenlijk afwijkt van het eenvoudige voorbeeldje dat in figuur 3/6.19-1 werd getekend. Een aantal ingangspoorten (links) is verbonden met een aantal uitgangspoorten (rechts). Tussen de uitgangen en de ingangen van deze poorten is een matrix

van verbindingslijnen opgenomen. Op ieder kruispunt van een horizontale en een vertikale lijn is een door te branden zekering opgenomen. Het al dan niet doorbranden van deze verbindingen programmeert de PLD voor het uitvoeren van de noodzakelijke logische functies. De bolletjes A op de kruispunten geven aan dat de zekering op die plaats nog aanwezig is en dat de horizontale lijn dus met de vertikale lijn is doorverbonden. Daar waar de lijnen elkaar snijden (B) wordt de zekering doorgebrand, zodat geen elektrische verbinding bestaat.

Voor- en nadelen van PLD's

De grote voordelen van het toepassen van PLD's zullen duidelijk zijn. In één complex PLD kunnen vaak logische functies geprogrammeerd worden, waar zonder PLD tientallen standaard TTL-IC's voor nodig zijn. Dat bespaart plaats op de print, ontwerptijd voor de print, montagetijd en controletijd. Bovendien staan tegenwoordig zoveel PLD's ter beschikking dat er steeds wel een schakeling gevonden kan worden, die zo economisch mogelijk geprogrammeerd kan worden voor een typische toepassing. In moderne PLD's zijn meer dan 6.000 AND- en OR-poorten aanwezig, zodat zelfs zeer complexe decodeerschakelingen in één IC aangebracht kunnen worden.

Een typisch voorbeeld van de enorme besparing aan onderdelen die gerealiseerd kan worden door het toepassen van dergelijke PLD's is getekend in figuur 3/6.19-4.



Figuur 3/6.19-4: Op de "Thunderbyte PC Immunizer" nemen twee PLD's de functie van tientallen normale IC's over.

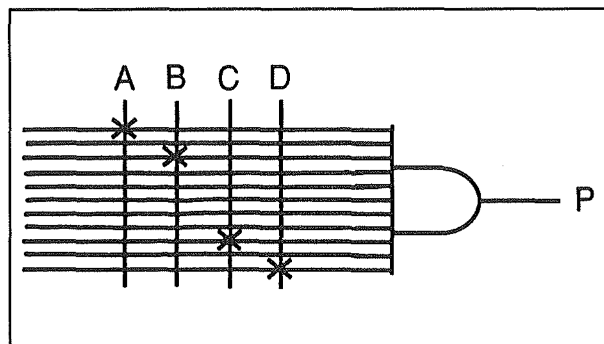
Op deze print, de "Thunderbyte PC Immunizer", worden twee PLD's toegepast waarin een zeer complexe decodeerlogica is geprogrammeerd. Dank zij deze twee IC's kan de zeer ingewikkelde schakeling op een zeer klein printje gerealiseerd worden!
Voor het toepassen van PLD's heeft men echter uiteraard specifieke kennis nodig. In ieder geval moet men zeer goed de Booleaanse algebra beheersen. Daarnaast moet men speciale apparatuur en software hebben, waarmee de logische functies in de PLD geprogrammeerd kunnen worden.

Herprogrammeerbare PLD's
De eerste generaties PLD's waren, net zoals PROM's, eenmalig programmeerbaar. Tegenwoordig bestaan er echter ook PLD's waarbij de programmatie door belichten met UV-licht of door het aanleggen van elektrische spanningen te wissen is! Men heeft hier dus dezelfde evolutie doorgemaakt als van de EPROM naar de EEPROM. Het zal duidelijk zijn dat hierdoor de praktische bruikbaarheid van deze schakelingen nog meer toeneemt. Een foutieve programmering, hetgeen zeker bij het programmeren van ingewikkelde schakelingen kan gebeuren, kan dus eenvoudig gecorrigeerd worden, zonder dat dit een IC tot gevolg heeft dat niet meer bruikbaar is.

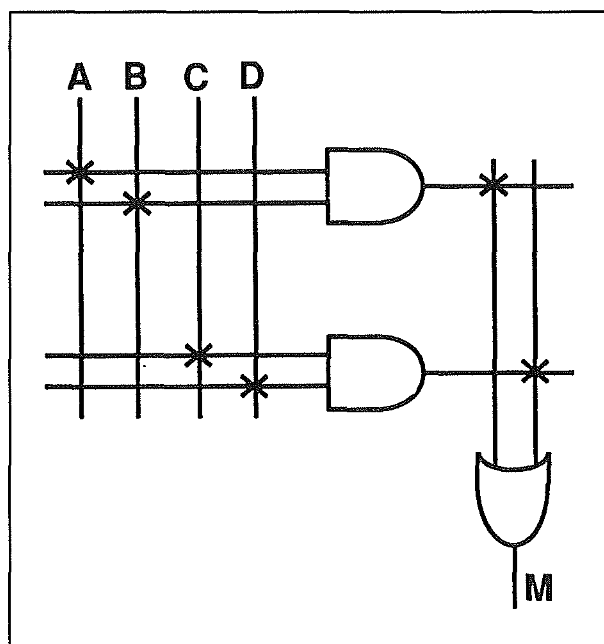
Theoretische achtergronden

Booleaanse algebra
PLD's worden, dat zal uit het voorbeeld wel duidelijk zijn geworden, gebruikt voor het oplossen van logische vergelijkingen, die in de "normale" elektronica met AND's, OR's, NAND's, NOR's en/of inverters worden samengesteld. Nu zijn PLD's, om de zaken niet te gecompliceerd te maken, meestal alleen voorzien van een heleboel AND- en OR-schakelingen. Omdat het echter onmogelijk is om logische vergelijkingen uit te voeren zonder inverters, worden alle ingangssignalen intern nog eens geïnverteerd en aan de lijnenmatrix aangeboden.
Het komt er nu op aan de standaard logische vergelijkingen om te zetten tot formules, waarin alleen AND-, OR- en inverter-functies voorkomen.
Dit kan met behulp van de Booleaanse algebra, die zich gespecialiseerd heeft in het oplossen van dergelijke logische vergelijkingen.
Een AND-functie wordt in die algebra voorgesteld door de uitdrukking:
P = A * B * C * D
Het sterretje geeft aan dat er een AND-relatie bestaat tussen de operatoren A, B, C en D. In normale mensentaal kan men deze formule interpreteren als *"P is waar als én A én B én C én D waar zijn"*. In elektronisch jargon kan men stellen dat de grootheid P alleen "H" is als én A én B én C én D "H" zijn.
Een dergelijke AND-vergelijking noemt men ook wel eens een "product-term".
Een OR-functie wordt in de Booleaanse algebra uitgedrukt door de vergelijking:
P = A + B + C + D
Het plusje geeft de OR-relatie aan tussen de operatoren A, B, C en D. Onder woorden gebracht betekent deze formule: *"P is waar als óf A óf B óf C óf D waar is "*. Het signaal H wordt "H" als een van de signalen A, B, C óf D "H" is.
De OR-formule wordt ook vaak de "somterm" genoemd.

6.19 Werking en principes van PLD's



Figuur 3/6.19-5: Het programmeren van een product-term in de matrix van een PLD.



Figuur 3/6.19-6: Het programmeren van een som-functie.

Een voorbeeld

Dergelijke logische AND/OR-vergelijkingen komt men vaak tegen in de procesindustrie. Een voorbeeld.
In een chemische fabriek mag bijvoorbeeld de kraan KRAAN_1 in een leiding alleen open gedraaid worden als de temperatuur TEMP_1 in een reactievat goed is en als een bepaalde procestijd TIJD_2 verstreken is. Maar dezelfde kraan moet ook opengedraaid worden als de druk DRUK_MAX in het reactievat te hoog wordt en als een bepaald alarmsignaal SIGNAAL_5 niet aanwezig is.
Door gebruik te maken van de twee basisformules van de Booleaanse algebra kan men dit hele mondvol samenvatten tot:
KRAAN_1 = [TEMP_1 * TIJD_2] + [DRUK_MAX * $\overline{\text{SIGNAAL_5}}$]
Het streepje boven SIGNAAL_5 is de negatie van de voorwaarde, immers KRAAN_1 mag alleen open gaan als dat signaal NIET aanwezig is! Een negatie wordt in de Booleaanse algebra voorgesteld door een streepje boven de operator te trekken.
Een dergelijke Booleaanse formule kan nu zonder problemen in een PLD geprogrammeerd worden, die alleen bestaat uit AND- en OR-poorten! Door de interne inversie van alle ingangssignalen heeft men immers ook de beschikking over het signaal $\overline{\text{SIGNAAL_5}}$.

De AND-functie programmeren

Bij alle PLD's worden de AND-poorten verbonden met de ingangsmatrix van het IC. Hoe de Booleaanse vergelijking:
P = A * B * C * D
in een dergelijke matrix te programmeren is, is getekend in figuur 3/6.19-5.

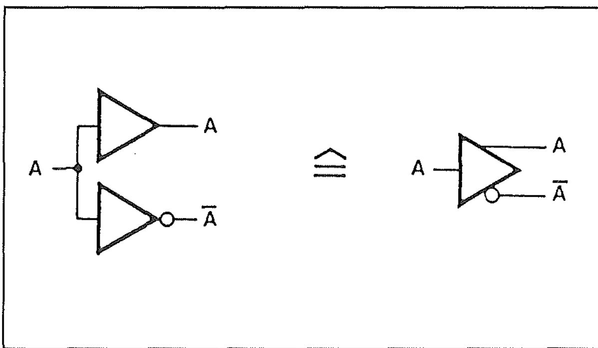
Alleen de zekeringen op de matrixverbindingen tussen de ingangen A, B, C en D en vier willekeurige ingangen van een AND in de PLD worden niet opgeblazen. Door alle overige zekeringen wel door te smelten wordt de AND-functie in het IC geprogrammeerd.

De OR-functie programmeren

Bij alle PLD's staan de OR-poorten in de uitgangsstructuur van het IC. Met moet de logische vergelijking dus eerst in product-

termen samenvatten, die nadien samengevoegd kunnen worden tot som-termen. De OR-functie:
M = [A * B] + [C * D]
kan volgens figuur 3/6.19-6 in de uitgangsstructuur van een PLD geprogrammeerd worden.

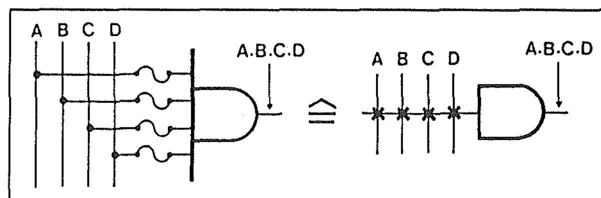
De ingangsstructuur van een PLD
Zoals reeds opgemerkt is het meestal onmogelijk logische functies te omschrijven met alleen product- en som-termen als men niet de beschikking heeft over de inverse waarde van operatoren. Vandaar dat alle praktische PLD's de ingangen bufferen met schakelingen, die ook de geïnverteerde waarde van de ingangen genereren. De ingangsstructuur van een PLD ziet er dan ook meestal uit zoals getekend in figuur 3/6.19-7.



Figuur 3/6.19-7: De ingangsstructuur van een PLD.

Een ingangssignaal A wordt aangeboden aan twee buffers, waarvan er een niet-inverterend en een wel-inverterend werkt. De twee uitgangssignalen A en $\overline{A}$ gaan dan naar de lijnen van de matrix. In de interne schema's van PLD's wordt deze typische ingangsstructuur voorgesteld door het rechts in de figuur getekende symbooltje. De twee parallel werkende buffers zorgen ervoor dat er geen signaalvertragingen ontstaan tussen de signalen A en $\overline{A}$ die naar de matrix gestuurd worden.

De AND- en OR-structuren
In figuur 3/6.19-5 werden alle ingangen van de AND-poort in de matrix ingetekend. In de praktijk doet men dat niet, omdat het schema dan veel te onoverzichtelijk wordt. Een AND-poort met meerdere ingangen (in dit geval vier) wordt voorgesteld door het rechter symbool in figuur 3/6.19-8.



Figuur 3/6.19-8: De voorstelling van een AND-poort met vier ingangen in het schema van een PLD.

De poort heeft dus slechts één horizontale ingangslijn in de matrix. Door de aanwezigheid van vier kruisjes op deze lijn weet de ontwerper dat de poort vier ingangen heeft.

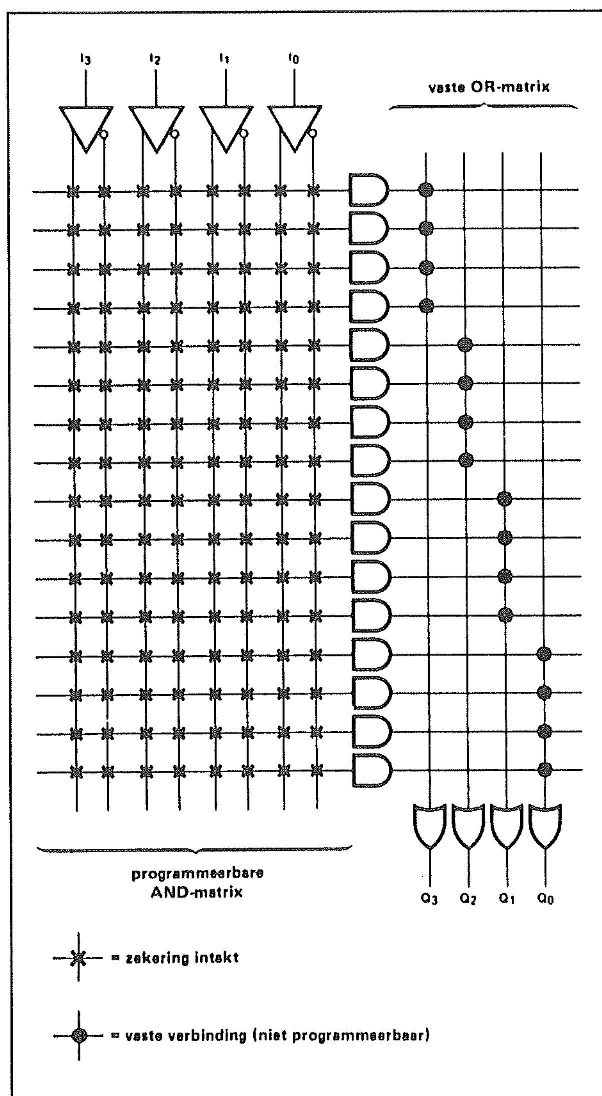
De PAL

Inleiding
PAL is het letterwoord van "Programmable Array Logic". Dit soort schakelingen worden gekarakteriseerd door een programmeerbare AND-logica aan de ingangen en een ***vaste*** structuur aan de OR-uitgangen. De uitgangen van de AND-poorten zijn dus vast verbonden met de ingangen van de OR-poorten. Deze indeling is niet zo vreemd als het lijkt. Met eenvoudige Booleaanse rekentechnieken

6.19 Werking en principes van PLD's

is het mogelijk gelijk welke logische vergelijking te schrijven als de som van product-termen. De product-termen worden geprogrammeerd in de ingangsmatrix van de AND-poorten, de som-termen hoeven dan niet meer geprogrammeerd te worden.

Typische structuur van een PAL
In figuur 3/6.19-9 is de typische structuur van een PAL van de eerste generatie getekend.



Figuur 3/6.19-9: De standaard structuur van een PAL van de eerste generatie.

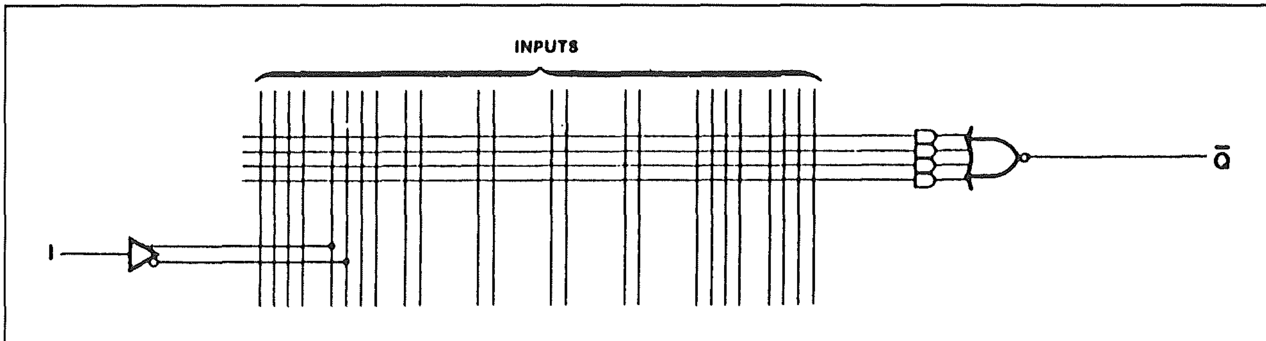
In het voorbeeld heeft de PAL vier ingangen en vier uitgangen.
De uitgangen van de 16 AND-poorten zijn, in groepjes van vier, vast verbonden met de ingangen van de vier OR-poorten aan de uitgangen.

Soorten PAL's
De eenvoudige opzet, waarbij een PAL alleen bestaat uit AND-poorten aan de ingangen en OR-poorten aan de uitgangen, is niet lang gehandhaafd. Tegenwoordig zijn er ook PAL's verkrijgbaar waarin:
- andere soorten poorten dan alleen AND en OR aanwezig zijn;
- flip-flop's (registers) in de uitgangen beschikbaar zijn;
- het mogelijk is signalen van de uitgang naar de ingang terug te koppelen;
- een clock-lijn aanwezig is voor het triggeren van de flip-flop's;
- een enable-lijn programmeerbaar is die uitgangen in tri-state kan schakelen.

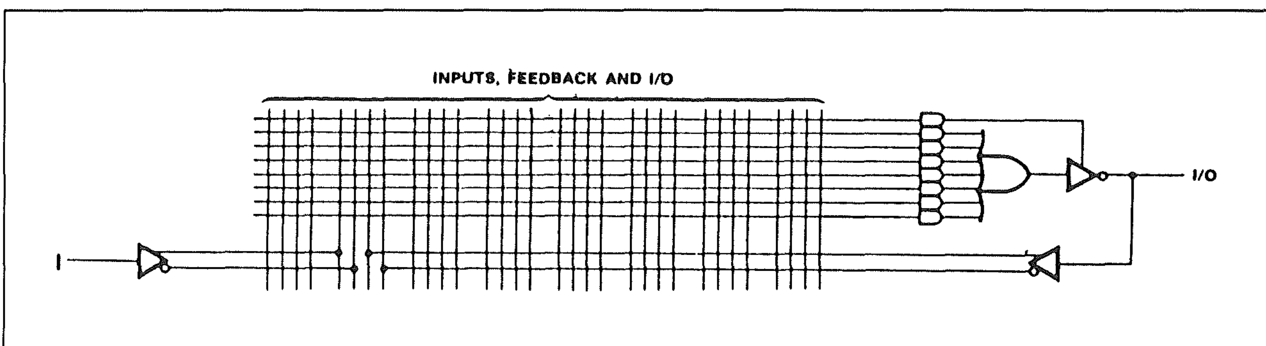
Een en ander heeft tot gevolg dat het niet eenvoudig is om voor een bepaalde opdracht de juiste schakeling uit te kiezen!

Voorbeelden
In de figuren 3/6.19-10 tot en met -14 zijn enige voorbeelden van dergelijke geavanceerde PAL's getekend. Voor de duidelijkheid is steeds maar één ingang getekend, die naar één uitgang gaat.
- Figuur 3/6.19-10:
  In deze PAL wordt gebruik gemaakt van een NOR-poort aan de uitgang. Vandaar de inverse notering $\overline{Q}$ aan de uitgang. Dergelijke PAL's wordt AND/NOR genoemd.
- Figuur 3/6.19-11:
  Bij deze PAL wordt de uitgang terug gekoppeld naar de ingangsmatrix.

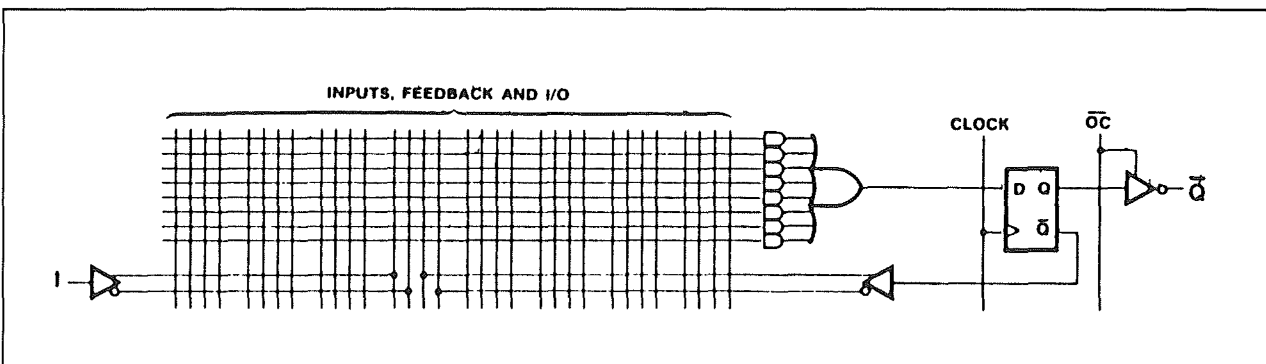
8.9 Software voor de ontwerper



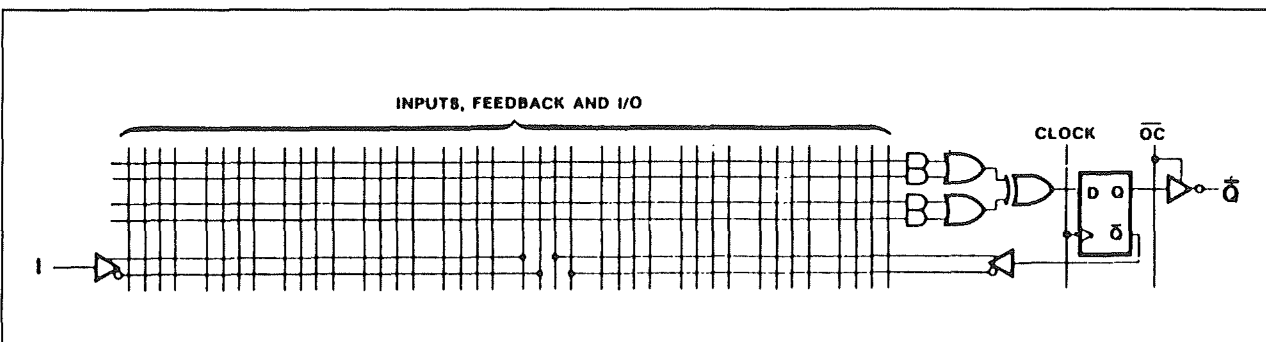
Figuur 3/6.19-10: Een PAL van het type L.



Figuur 3/6.19-11: Een PAL van het type L+I/O.

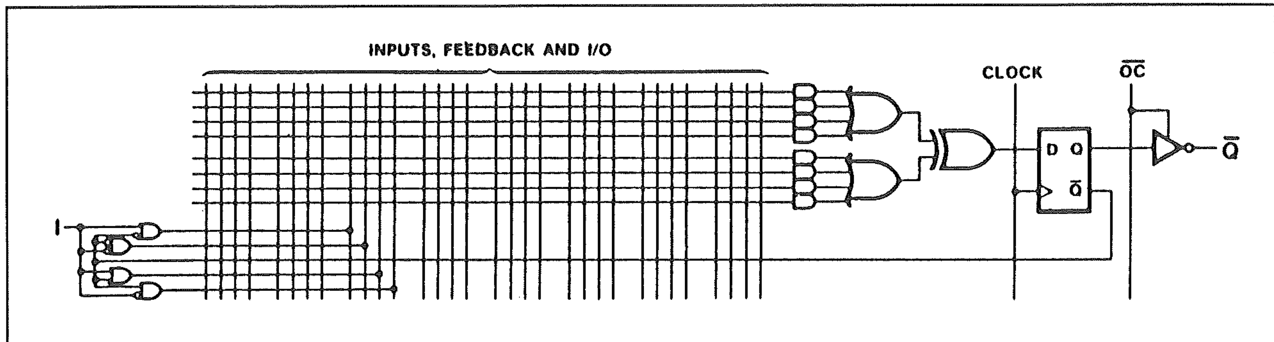


Figuur 3/6.19-12: Een PAL van het type R.



Figuur 3/6.19-13: Een PAL van het type X.

6.19 Werking en principes van PLD's



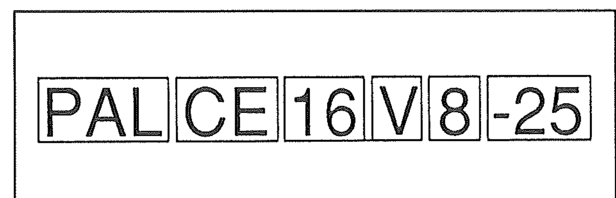
Figuur 3/6.19-14: Een PAL van het type A.

Op deze manier wordt deze uitgang ook tot ingang bevorderd, vandaar de notatie I/O bij de uitgangspen. Op deze manier kan men in de schakeling schuifregisters programmeren, of allerlei soorten eenvoudige geheugens.

- Figuur 3/6.19-12:
  Bij deze PAL worden de uitgangen van de NOR-poorten aangesloten op de D-ingangen van type-D flip-flop's. Alle flip-flop's worden gestuurd uit een clock-lijn. De Q-uitgangen van de flip-flop's gaan naar uitgangspoorten, die een tri-state besturing $\overline{OC}$ hebben. De Q-uitgangen worden bovendien terug gekoppeld naar de ingangsmatrix. Het zal duidelijk zijn dat men met dergelijke schakelingen tamelijk ingewikkelde logische vergelijkingen kan samenstellen.
- Figuur 3/6.19-13:
  Een identieke structuur, maar nu staat er tussen iedere NOR-uitgang en D-ingang nog eens een EXOR-poort.
- Figuur 3/6.19-14:
  De $\overline{Q}$-uitgang van de flip-flop wordt teruggekoppeld naar een poort-netwerk aan de ingang.
  Deze uitgang kan nu opgenomen worden in eenvoudige logische somtermen met de ingang I, zoals [I + Q], [I + $\overline{Q}$], etc.

Codering van PAL's

Men heeft pogingen gewaagd om een codering op te stellen, waarbij uit het typenummer iets valt af te lezen over de interne samenstelling van de schakeling.
De standaard codering van PAL's is getekend in figuur 3/6.19-15.



Figuur 3/6.19-15: De standaard codering van PAL's, die door de belangrijkste fabrikanten wordt gevolgd.

Een toelichting op de samenstelling van het typenummer:

- Blok 1:
  Bevat meestal de letters PAL, als identificatie van het soort schakeling of een ander letterwoord waarmee de fabrikant "zijn" PAL's benoemd.
- Blok 2:
  De letter C duidt op de CMOS-technologie, waarmee de schakeling gemaakt is. De tweede letter E geeft aan dat de programmering weer elektrisch ongedaan kan worden gemaakt.
- Blok 3:

Het getal in dit blok geeft het aantal ingangen van de matrix aan, in dit geval dus 16.

- Blok 4:
  De letter in dit blok geeft aan hoe de schakeling in de PAL is samengesteld:
  - H: AND/OR, zoals bij de allereerste PAL's;
  - L: AND/NOR, zie figuur 3/6.19-10;
  - C: AND/OR plus AND/NOR;
  - R: flip-flop in de uitgang, zie figuur 3/6.19-12;
  - X: AND/NOR/EXOR plus flip-flop, zie figuur 3/6.19-13;
  - A: uitgang teruggekoppeld naar ingangspoorten met het zogenoemde "Borrow-bit", zie figuur 3/6.19-14;
  - P: programmeerbare polariteit van de uitgangen;
  - V: de OR-poorten aan de uitgang van de matrix hebben niet allemaal even veel ingangen.
- Blok 5:
  Het getal in dit blok geeft het aantal uitgangen aan, in dit geval dus acht.
- Blok 6:
  Het getal achter het streepje geeft de looptijd van de schakeling aan, dus de tijd in ns die een signaal nodig heeft om van de ingang naar de uitgang te migreren. In dit geval bedraagt de looptijd dus 25 ns.

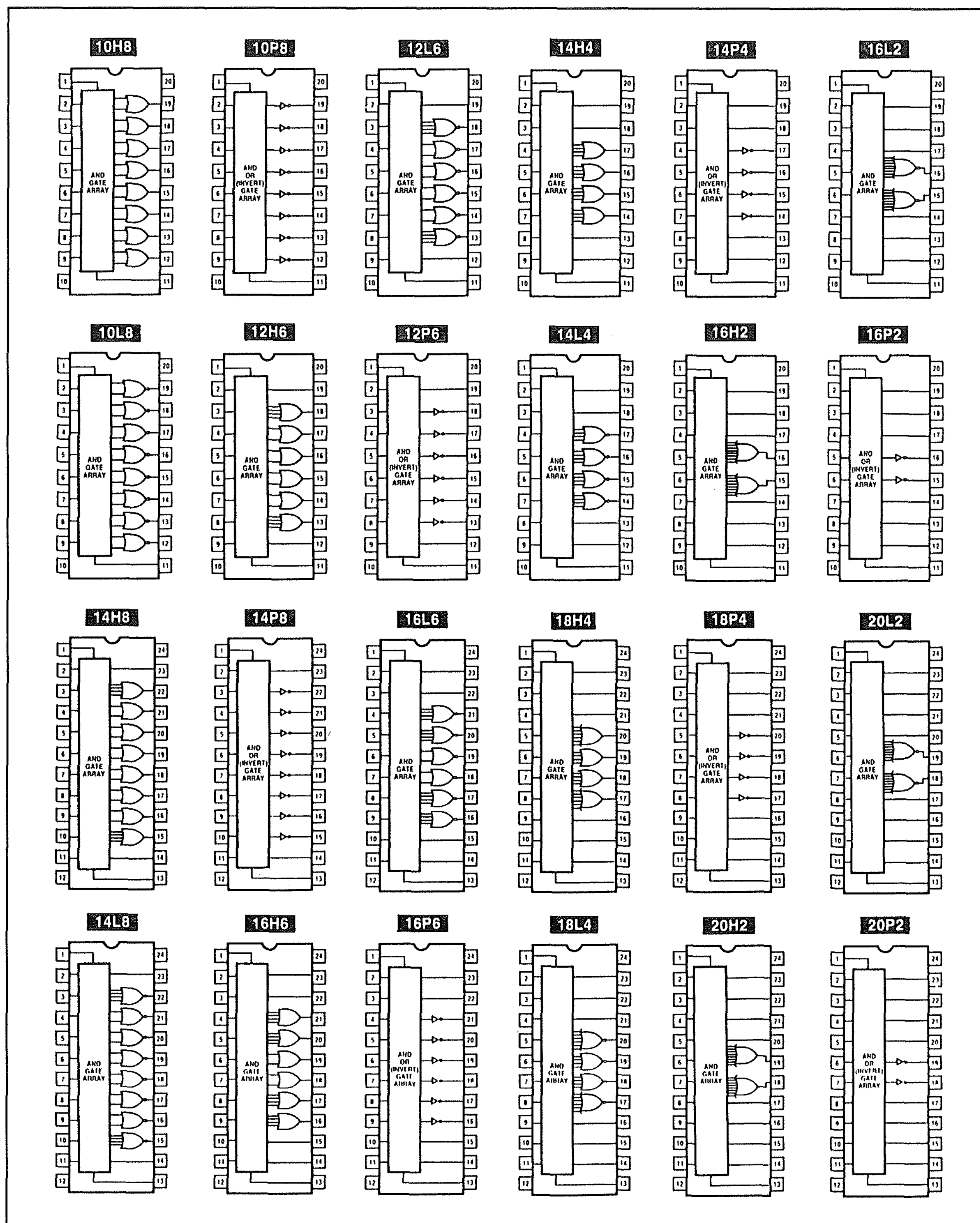
Bij de definitie van het aantal in- en uitgangen in het typenummer wordt alleen aangegeven hoeveel in- en uitgangen aan de functie-definitie voldoen. Zo heeft de PAL16R4 acht uitgangen, maar slechts vier die een type-D flip-flop hebben.

Overzicht van de meest gebruikte PAL's
In de tabel van figuur 3/6.19-16 is een overzicht gegeven van de eigenschappen van de meest toegepaste PAL's.

PAL	ingangen (I)	uitgangen (Q)	ingangen/ uitgangen (I/O)	registers	funktie
10H8	10	8			AND-OR
12H6	12	6			AND-OR
14H4	14	4			AND-OR
16H2	16	2			AND-OR
10L8	10	8			AND-OR-INVERT
12L6	12	6			AND-OR-INVERT
14L4	14	4			AND-OR-INVERT
16L2	16	2			AND-OR-INVERT
16C1	16	1			AND-OR/AND-OR-INVERT
16L8	10	8	6		AND-OR-INVERT
16R8	8	8			AND-OR-INVERT-REGISTER
16R6	8	8	2	6	AND-OR-INVERT-REGISTER
16R4	8	8	4	4	AND-OR-INVERT-REGISTER
16X4	8	8	4	4	AND-OR-INVERT-XOR-REGISTER
16A4	8	8	4	4	AND-CARRY-OR-XOR-INVERT-REGISTER

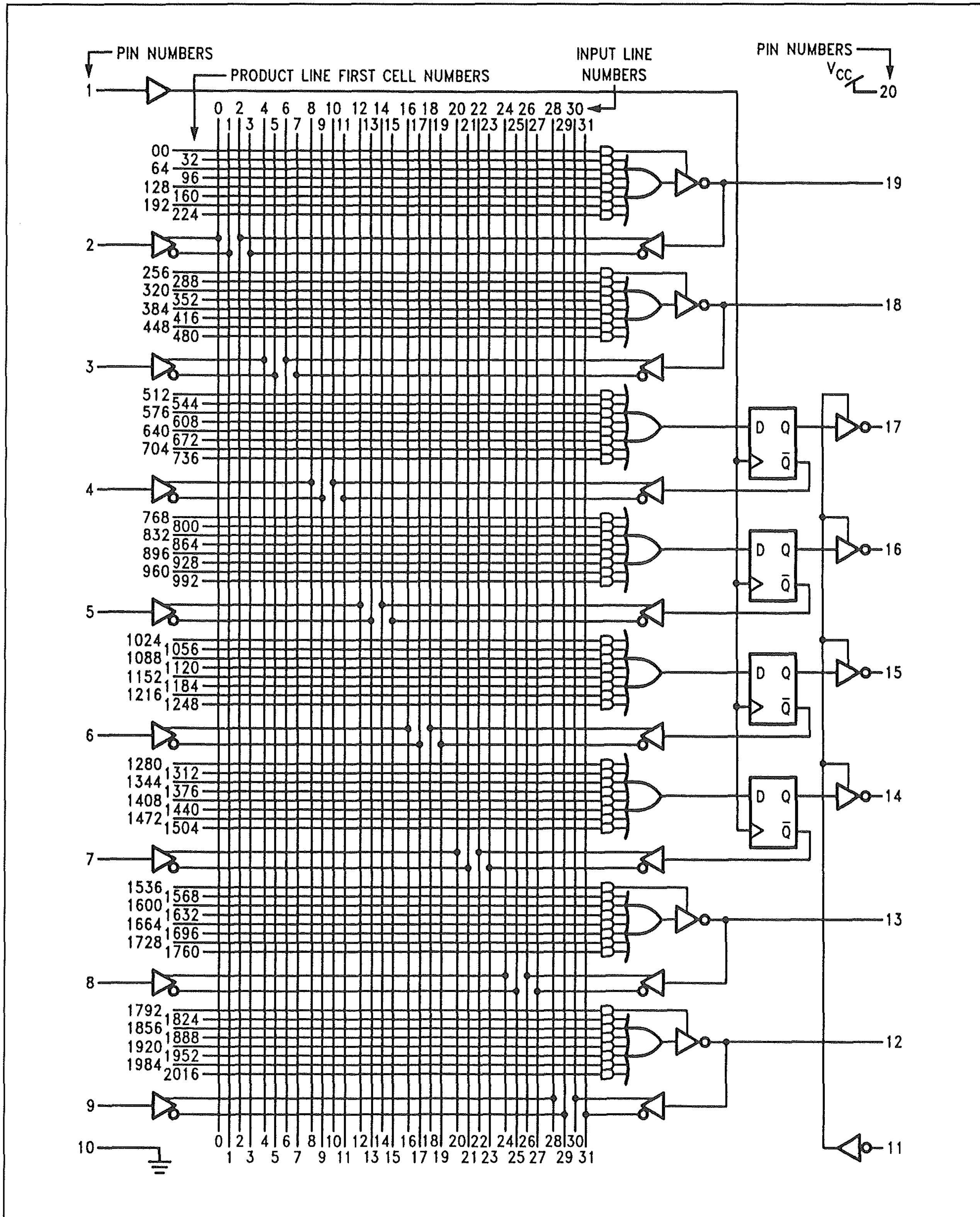
Figuur 3/6.19-16: Overzicht van de belangrijkste eigenschappen van veel gebruikte PAL's.

6.19 Werking en principes van PLD's



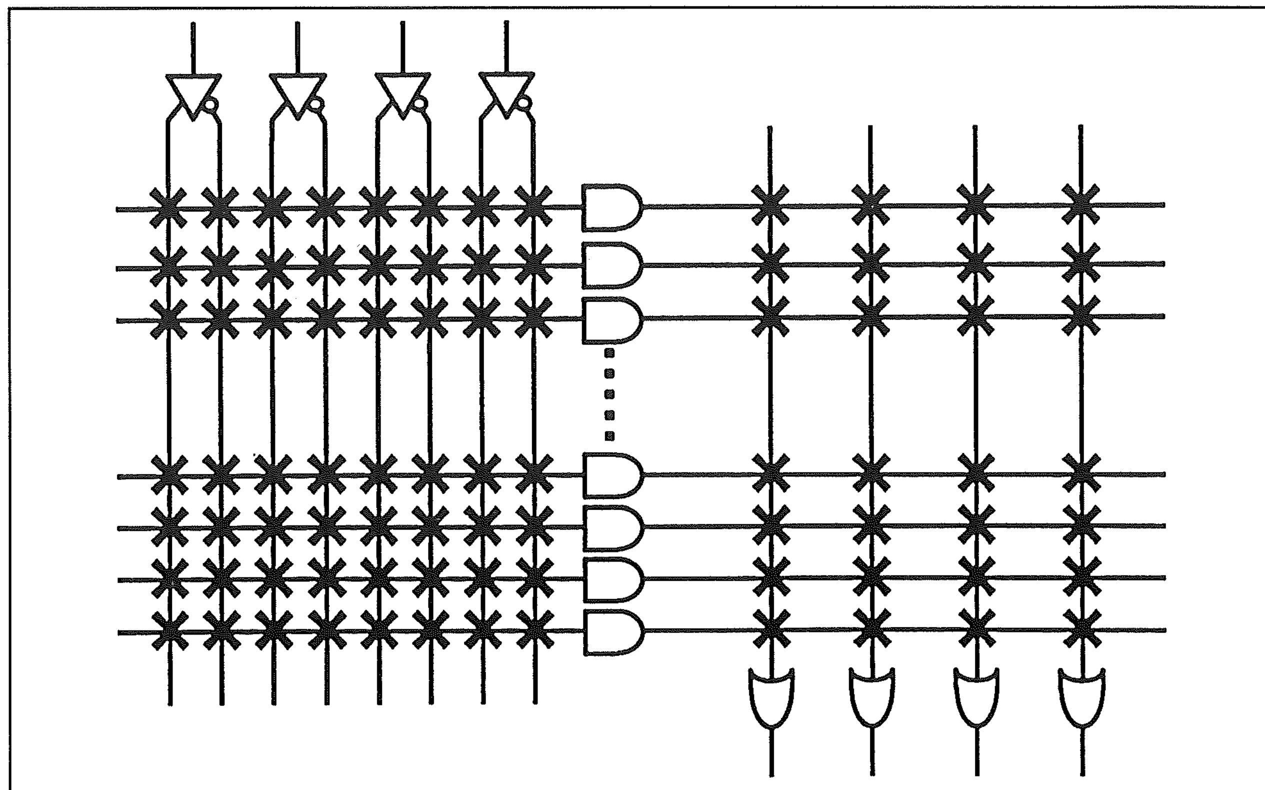
Figuur 3/6.19-17: De aansluitgegevens van 24 veel toegepaste PAL's.

8.9 Software voor de ontwerper



Figuur 3/6.19-18: Het interne schema van de PAL 16R4.

6.19 Werking en principes van PLD's



Figuur 3/6.19-19: De interne structuur van een PLA.

Een voorbeeld

In figuur 3/6.19-18 wordt tot slot van de bespreking van de PAL's het interne schema getekend van de PAL 16R4.

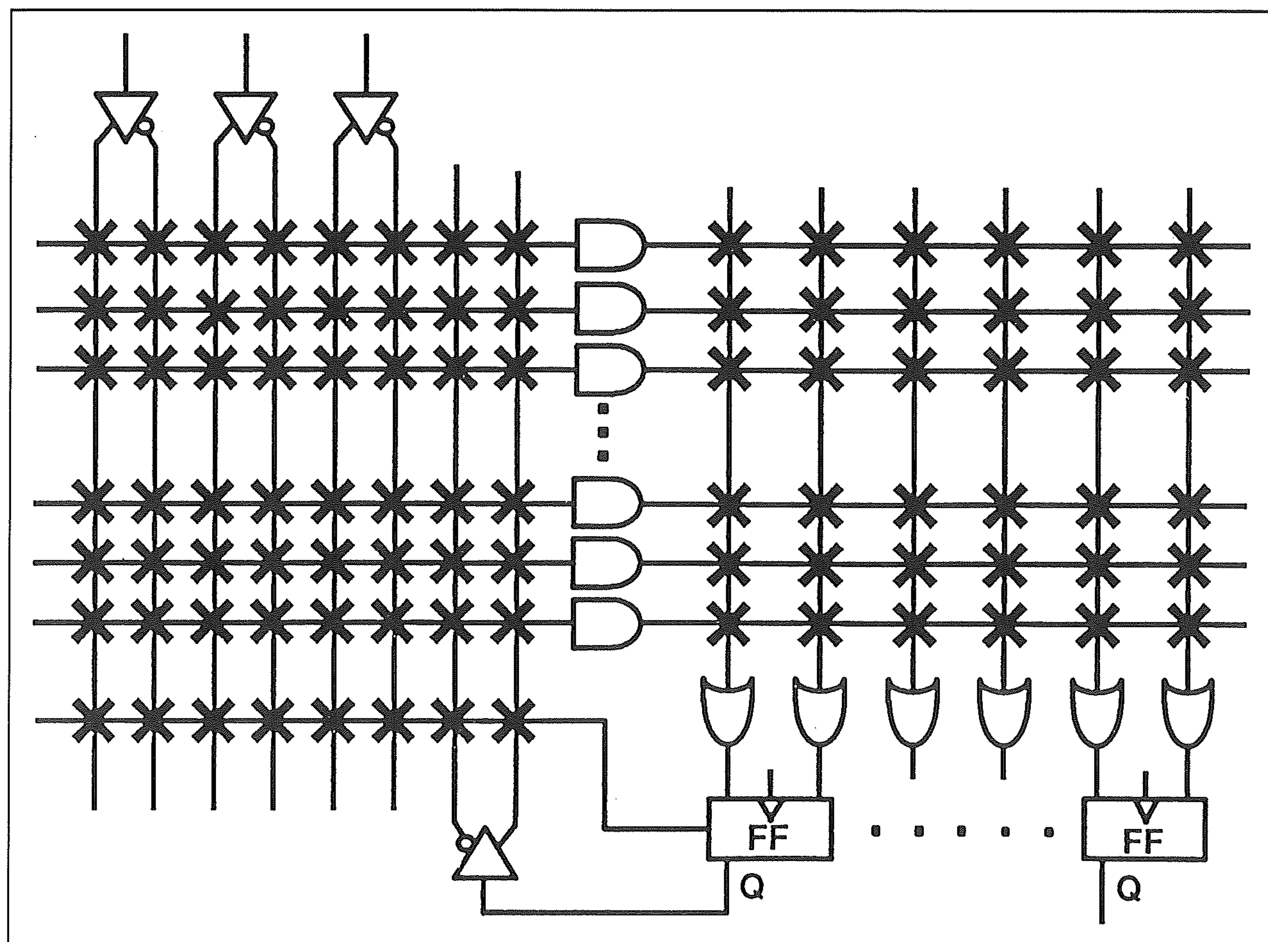
Snijpunt codering

Alle horizontale en vertikale lijnen van de matrix hebben een codering. Deze codering wordt gebruikt voor het identificeren van het snijpunt waarop een zekering is aangebracht.
Deze codering is heel belangrijk, want deze wordt opgenomen in de JEDEC-file die naar de PLA-programmer wordt gestuurd.
De codering van een snijpunt, volgens de internationaal gestandaardiseerde JEDEC-norm, is gelijk aan de som van het rij- en kolomnummer.
Of, in meer technische terminologie: de code van een snijpunt is gelijk aan het "Product Line First Cel Number" (horizontale lijn) plus het "Input Line Number" (vertikale lijn).

De PLA

Inleiding

PLA is het letterwoord voor "Programmable Logic Array".
De basisstructuur van een PLA is gelijk aan deze van een PAL, met als enig verschil dat nu ook de uitgangsmatrix volledig vrij te programmeren is. Een typische PLA van de eerste generatie is getekend in figuur 3/6.19-19.



Figuur 3/6.19-20: De basisstructuur van een PLS.

Iedere product-term kan dus aan iedere OR toegevoerd worden. Hierdoor neemt uiteraard de vrijheid van de programmeur toe, met als nadelige consequenties een groter stroomverbruik en geringere snelheid. Bovendien vereist het programmeren van PLA's veel meer programmeerkennis dan voor PAL's.

Opmerking

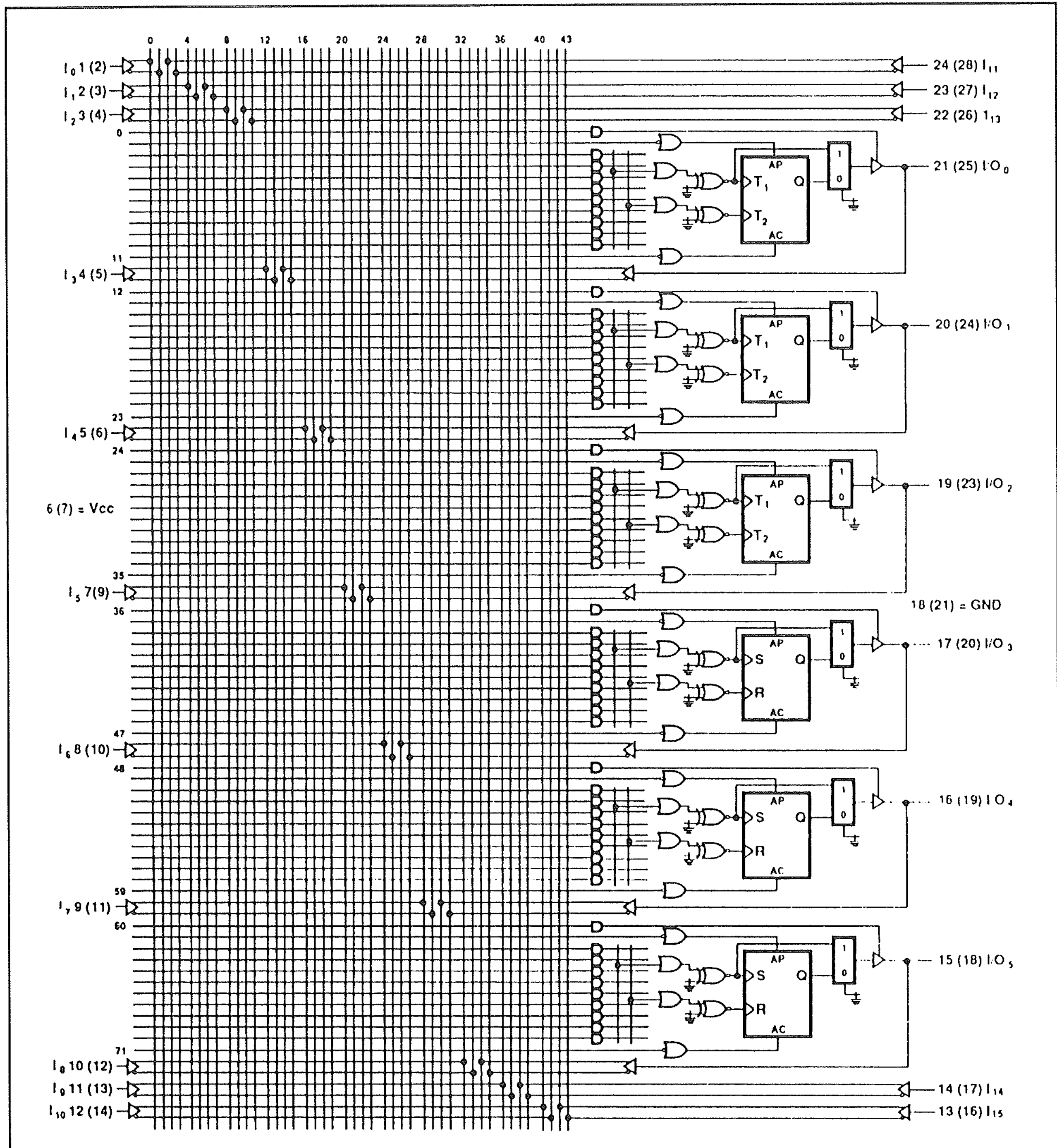
PLA's worden door sommige fabrikanten FPLA's genoemd. De F staat dan voor "Field".

De PLS

Inleiding

PLS is het letterwoord voor "Programmable Logic Sequencer". Een PLS is een PLA, dus met programmeerbare in- en uitgangsmatrix, maar met de toevoeging van flip-flop's aan de uitgang. Bovendien worden de uitgangen van deze flip-flop's in ieder geval terug gekoppeld naar de ingangen. De basisstructuur van een PLS is getekend in figuur 3/6.19-20.

6.19 Werking en principes van PLD's



Figuur 3/6.19-21: Het interne schema van een PLS, de 22IP6 van AMD.

In feite is het enige verschil tussen een PLS en een PAL van het R-type het feit dat ook de uitgangen programmeerbaar zijn. De voornaamste eigenschap van een PLS is dat de toestand op de uitgangen op een bepaald moment niet alleen meer afhankelijk is van de toestanden op de ingangen op dat moment, maar ook van de situatie zoals die voorheen was. In de registers (de flip-flop's) kan de uitgangssituatie voor

een clock-puls immers opgeslagen worden.

Voorbeeld
Een typisch voorbeeld van een PLS is de 22IP6 van AMD. Het intern schema van dit PLD is getekend in figuur 3/6.19-21.

Met stelt nu vast dat de uitgangen van de AND-poorten niet vast verbonden zijn met de ingangen van de OR-poorten, maar via programmeerbare matrixen.

De GAL

Inleiding
GAL is het letterwoord van "Generic Array Logic". Een GAL heeft dezelfde structuur als een PAL, maar in iedere uitgang is een zogenoemd OMC aanwezig. Dit is de afkorting van "Output Macro Cell". Zoals uit figuur 3/6.19-22 blijkt is de matrix in de ingangen programmeerbaar, maar bestaan er vaste verbindingen tussen de uitgangen van de AND-poorten en de ingangen van de OR-poorten. De OMC's hebben programmeerbare terugkoppelingen naar de ingangsmatrix.

Een andere eigenschap van GAL's is dat deze steeds herprogrammeerbaar zijn. Men kan de programmering weer wissen, waarna het IC beschikbaar is voor hernieuwde programmatie.

De Output Macro Cell
Het belangrijkste onderdeel van een GAL is de Output Macro Cell, OMC of soms ook OMLC genoemd. De samenstelling van een dergelijke cel in de GAL 16V8 van Lattice Semiconductor is getekend in figuur 3/6.19-23.

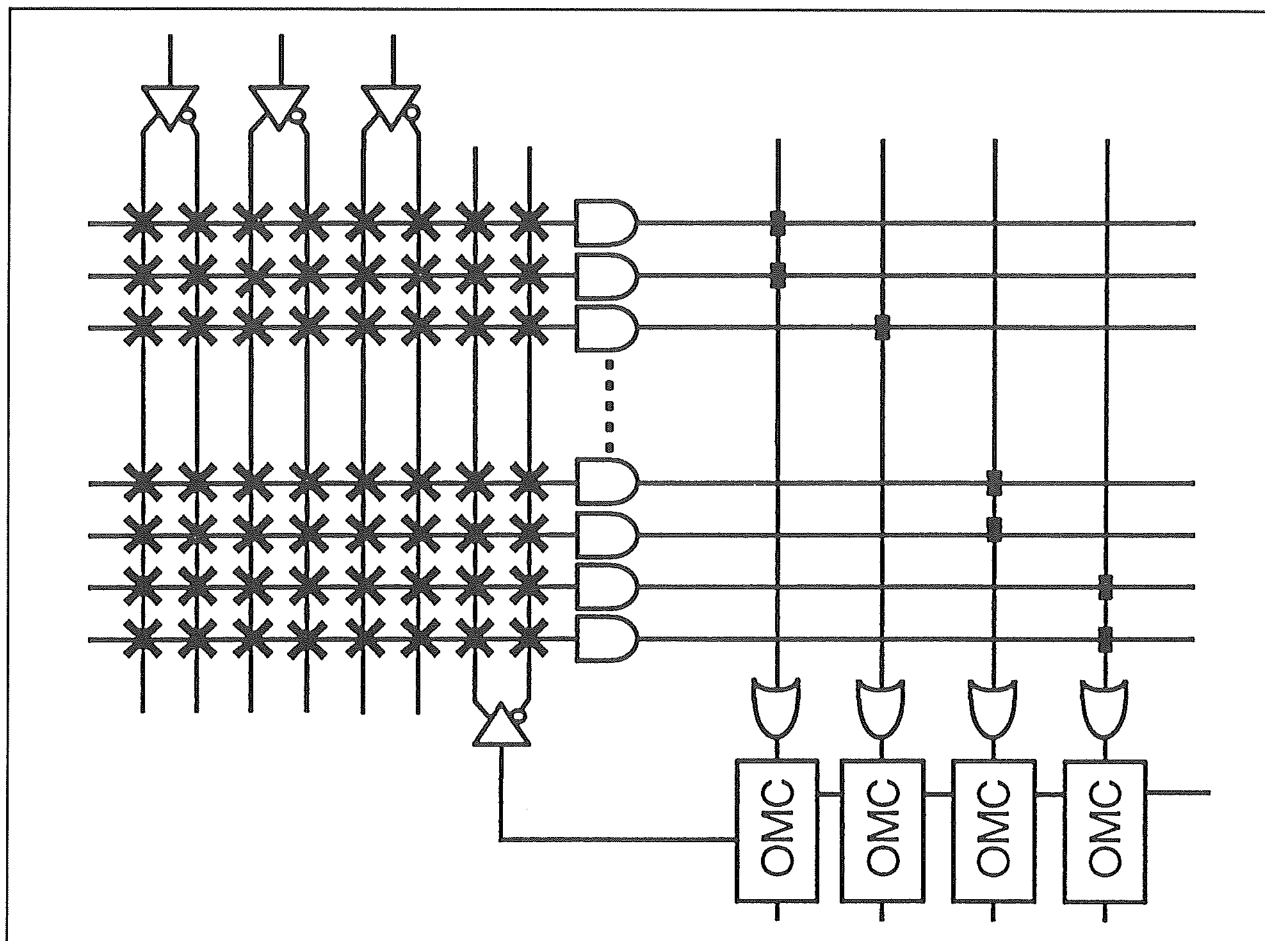
Merk op dat de OR-poort bij dit IC in de structuur van de OMC is opgenomen.
De OMC kan zo geprogrammeerd worden dat bijvoorbeeld:
- de uitgang inverterend werkt;
- de flip-flop in het data-pad wordt opgenomen;
- de uitgang teruggekoppeld wordt naar de ingangsmatrix.

Dank zij deze programmeerbare OMC is een GAL veel flexibeler dan andere typen PLD's.

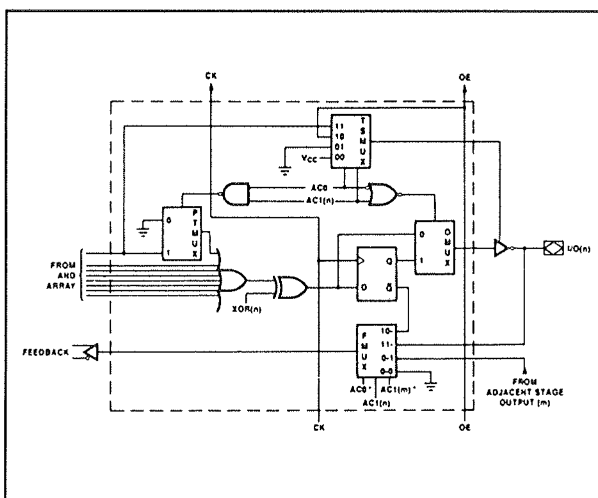
Hoe die cel geprogrammeerd wordt hangt uiteraard af van de fabrikant van het IC. Voor de reeds genoemde 16V8 van Lattice Semiconductor staan voor het programmeren drie bits ter beschikking:
- SYN;
- ACO;
- $AC_{(n)}$.

De SYN- en ACO-bits werken in op alle cellen, het $AC_{(n)}$-bit werkt per OMC.
In het kort worden enige mogelijke programmeringen besproken.
- Normale uitgangen:
  SYN=1; ACO=0; $AC_{(n)}$=0
  In dit geval is het net of de OMC niet aanwezig is, zie figuur 3/6.19-24. De uitgang van de OR-poort gaat via een EXOR naar de uitgang van het IC.
- Uitgang als ingang:
  SYN=1; ACO=0; $AC_{(n)}$=1
  Na deze programmering wordt de uitgang van de vorige trap als ingang gebruikt en teruggekoppeld naar de matrix in de ingang, zie figuur 3/6.19-25.
- Uitgang als I/O:
  SYN=1; ACO=1; $AC_{(n)}$=1
  De uitgangsbuffer levert nu wel een uitgang, maar deze wordt tevens teruggekoppeld naar de ingang, zie figuur 3/6.19-26.

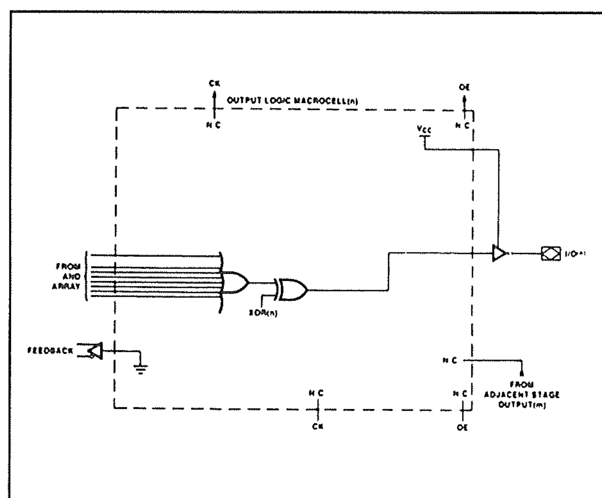
6.19 Werking en principes van PLD's



Figuur 3/6.19-22: Het basisschema van een GAL.

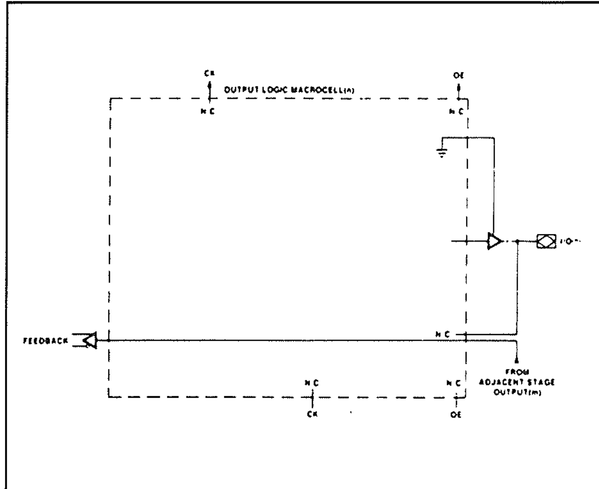


Figuur 3/6.19-23: De samenstelling van de OMC in de GAL 16V8 van Lattice Semiconductor.



Figuur 3/6.19-24: De OMC van een GAL geprogrammeerd als gewone uitgang.

8.9 Software voor de ontwerper



Figuur 3/6.19-25: De OMC geprogrammeerd als ingang.

- Uitgang met flip-flop:
  SYN=0; ACO=1; $AC_{(n)}$=0
  Nu wordt de flip-flop van de OMC in het data-pad van de uitgang (n) opgenomen, zie figuur 3/6.19-27. De Q-uitgang gaat via de uitgangsbuffer naar de uitgangspen, de $\overline{Q}$-uitgang wordt teruggekoppeld naar de ingangsmatrix.

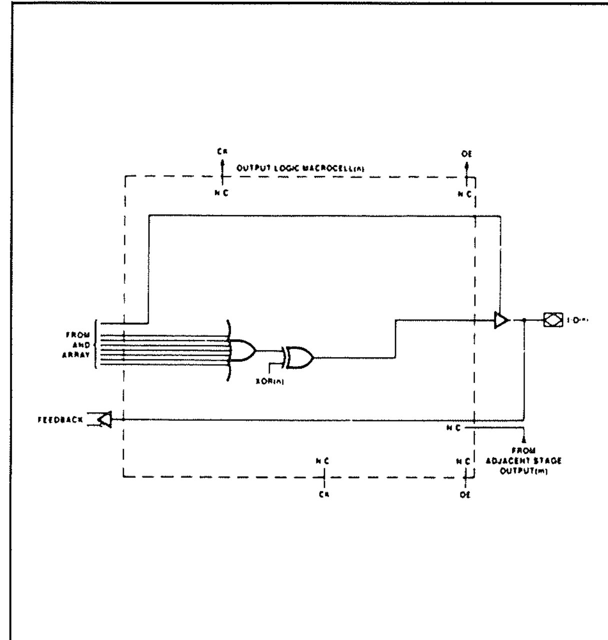
Besluit

Dank zij de programmeerbare OMC kan men zuiver softwarematig in één en dezelfde GAL alle structuren inlezen die men bij PAL's uit de type-codering moet afleiden.

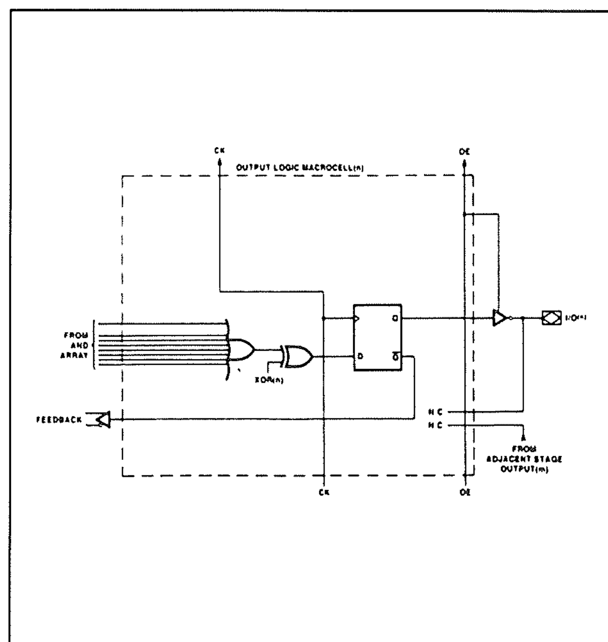
Met andere woorden, de schema's van de figuren 3/6.19-10 tot en met -14 kunnen allemaal in de OMC van een GAL geprogrammeerd worden. Met één IC kan men alle mogelijke structuren realiseren! GAL's zijn dus zonder meer de meest universele PLD's die op dit moment bestaan.

Een voorbeeld van een GAL

In figuur 3/6.19-28 is de interne structuur getekend van de GAL 16V8 van Lattice Semiconductor.



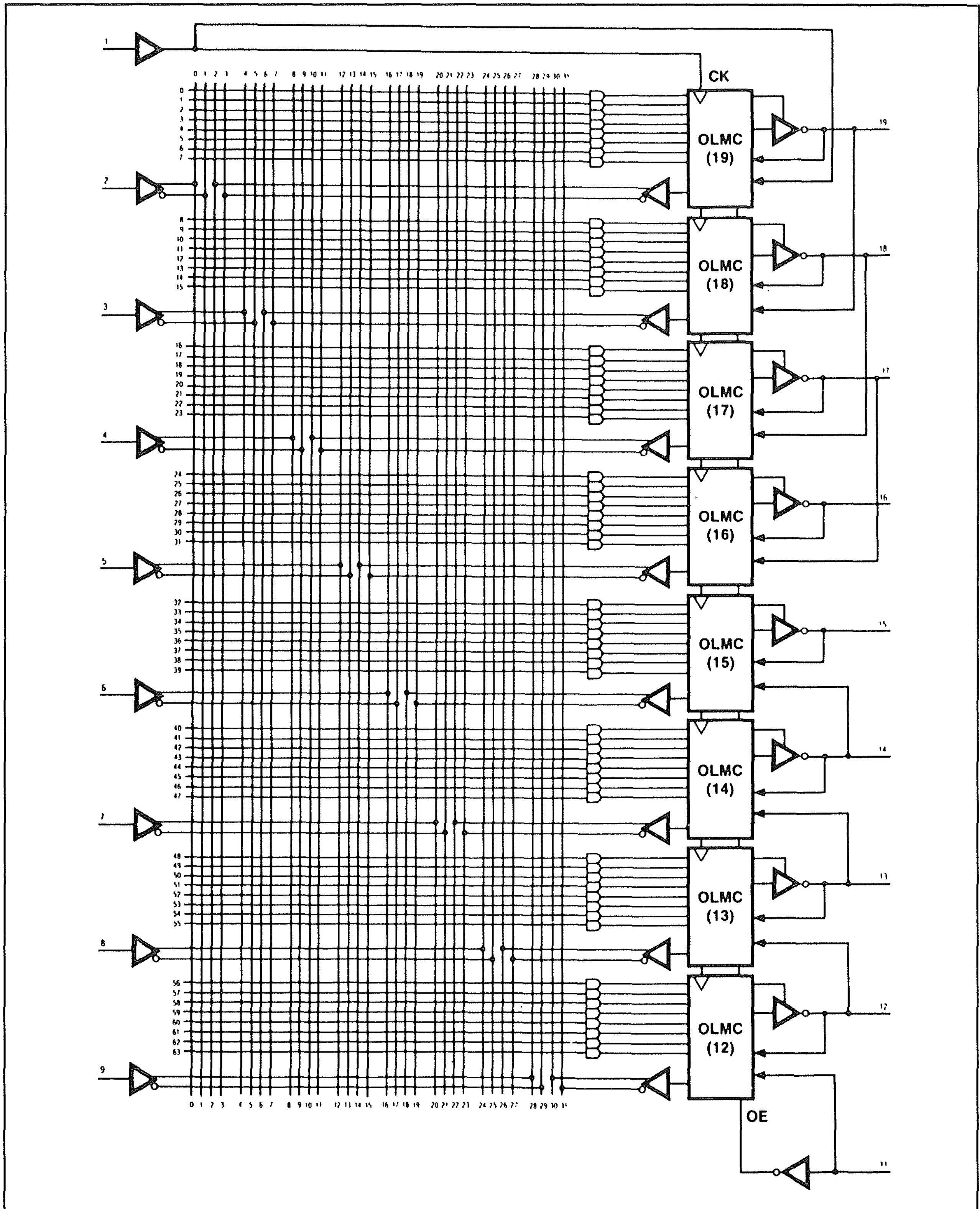
Figuur 3/6.19-26: De OMC geprogrammeerd als I/O.



Figuur 3/6.19-27: De OMC geprogrammeerd als register (flip-flop).

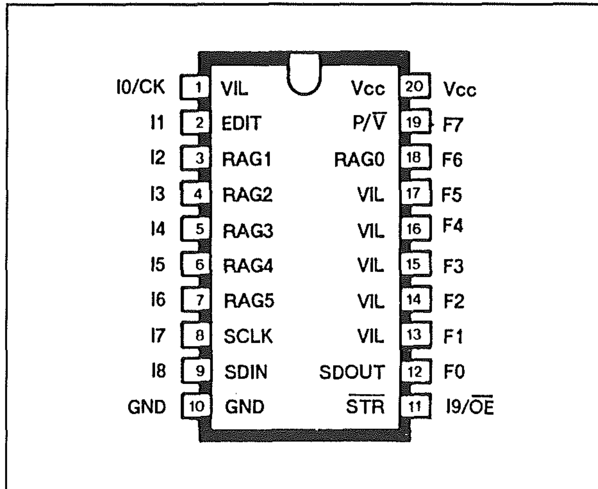
De aansluitgegevens van de 20-polige DIL-behuizing zijn getekend in figuur 3/6.19-29.

6.19 Werking en principes van PLD's



Figuur 3/6.19-28: Een voorbeeld van een GAL: de 16V8 van Lattice Semiconductor.

8.9 Software voor de ontwerper



Figuur 3/6.19-29: De aansluitgegevens van de GAL 16V8.

De zekeringen van PLD's

Inleiding

Alvorens de methoden te bespreken die ontwikkeld zijn voor het programmeren van PLD's moet uiteraard eerst het belangrijkste onderdeel bij dat programmeren besproken worden: de zekering, die op het snijpunt van een horizontale en vertikale lijn van de matrix aanwezig is.
In de loop der jaren zijn verschillende technologieën ontwikkeld, van de eenvoudige eenmalig te programmeren NiCr-zekering tot de meest ingewikkelde herprogrammeerbare halfgeleider structuren.

De NiCr-zekering

Deze zekering bestaat uit een dun laagje metallische legering, samengesteld uit chroom en nikkel.
Dit laagje is opgedampt tussen een horizontale en een vertikale lijn van de matrix, zie figuur 3/6.19-30. Door tussen de twee lijnen een spanning te zetten zal het dun laagje verdampen en wordt de verbinding tussen horizontale en vertikale lijn verbroken.

Per ingang zijn twee zekeringen noodzakelijk, een aan de I-kant en een aan de Ī-kant. In de getekende toestand is de zekering in de Ī-kant onderbroken, zodat de niet-geïnverteerde ingang met de ingang van een AND-poort wordt doorverbonden.
Het zal duidelijk zijn dat NiCr-zekeringen slechts één maal te programmeren zijn. Na het verdampen van het laagje kan de beginsituatie nooit meer hersteld worden.

De AIM-fuse

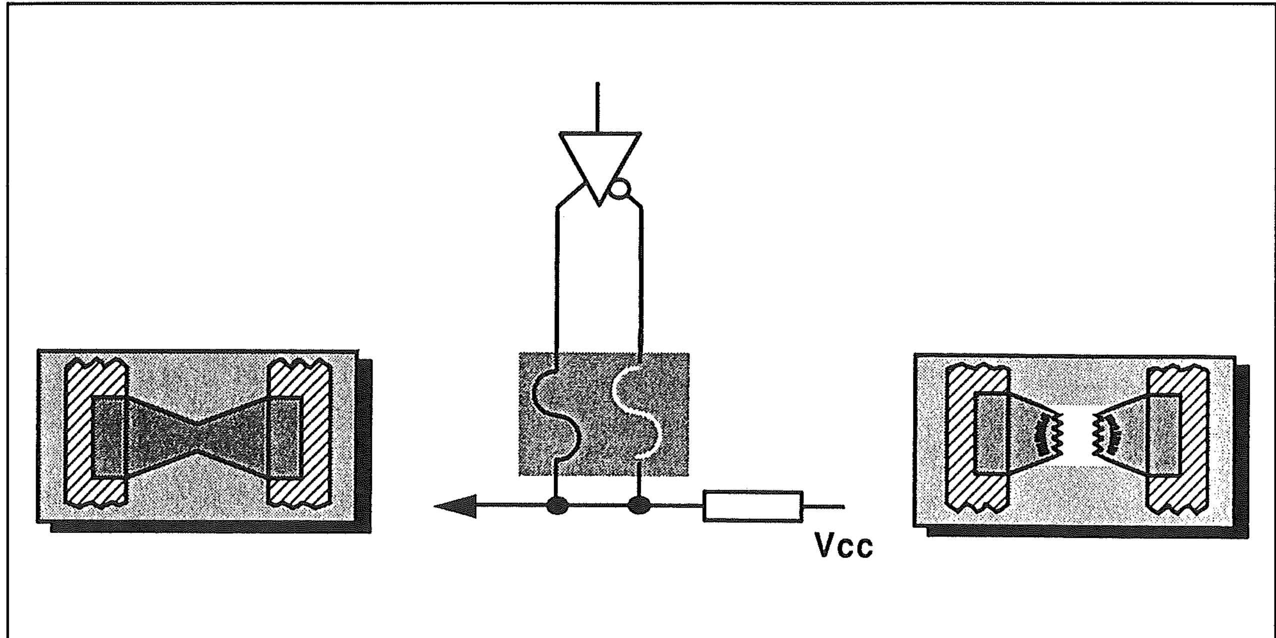
AIM is het letterwoord voor "Avelanche Induced Migration". Deze zekering, voorgesteld in figuur 3/6.19-31, is in feite een in het kristal van de chip vervaardigde diode.

Op een van de lagen van de diode is een dunne laag aluminium opgedampt. De diode wordt bij het programmeren kortgesloten, waardoor een geleidende verbinding ontstaat tussen de twee lijnen van de matrix. Het kortsluiten is een gevolg van het binnendringen van het aluminium in de N^+-laag van de diode. Hierdoor wordt deze laag overbrugd en de diode kortgesloten. Het grote verschil tussen de NiCr-zekeringen en de AIM-fuses is dat in het eerste geval een programmering een verbinding onderbreekt en in het tweede geval een verbinding maakt.

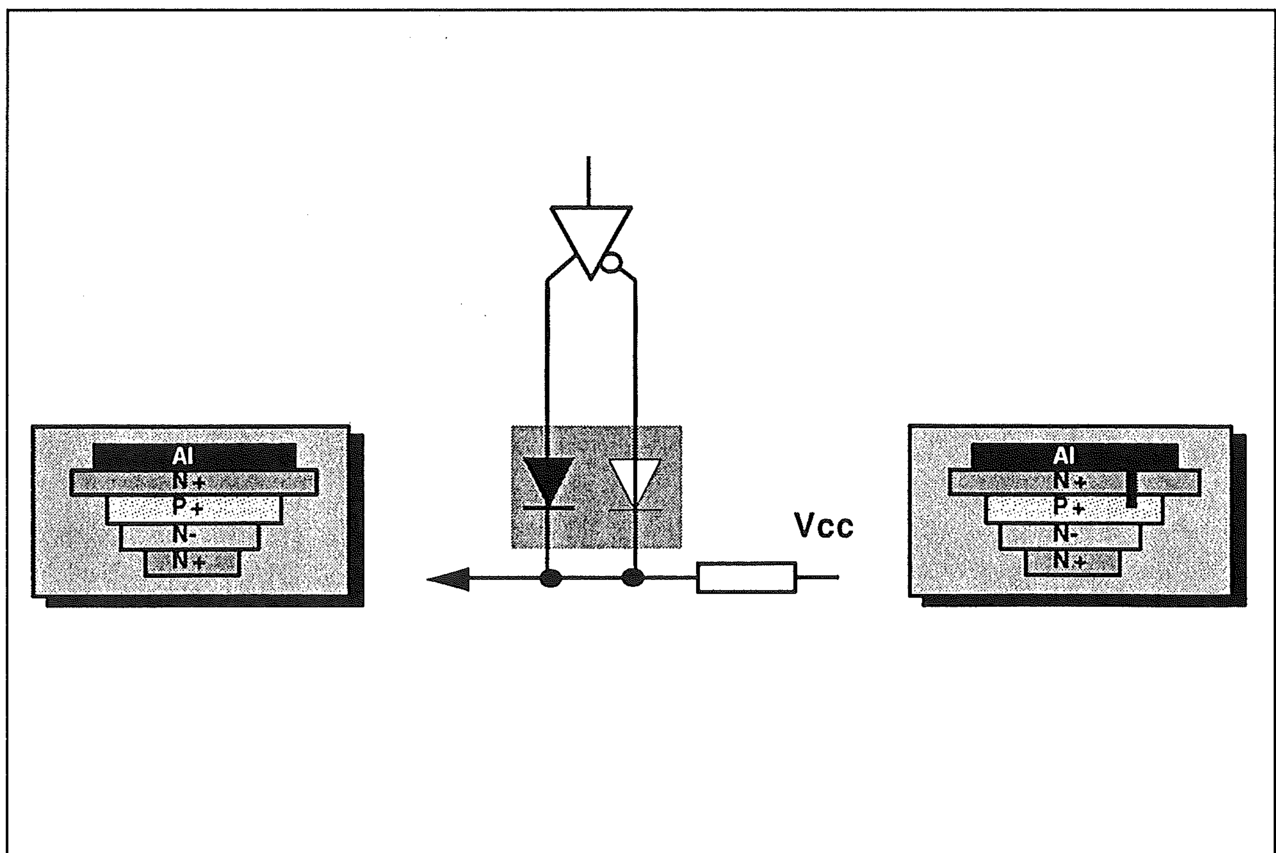
De CMOS-zekering

De moderne PLD's zijn allemaal uitgevoerd in CMOS-technologie. De zekering die bij deze technologie wordt gebruikt is getekend in figuur 3/6.19-32.

6.19 Werking en principes van PLD's

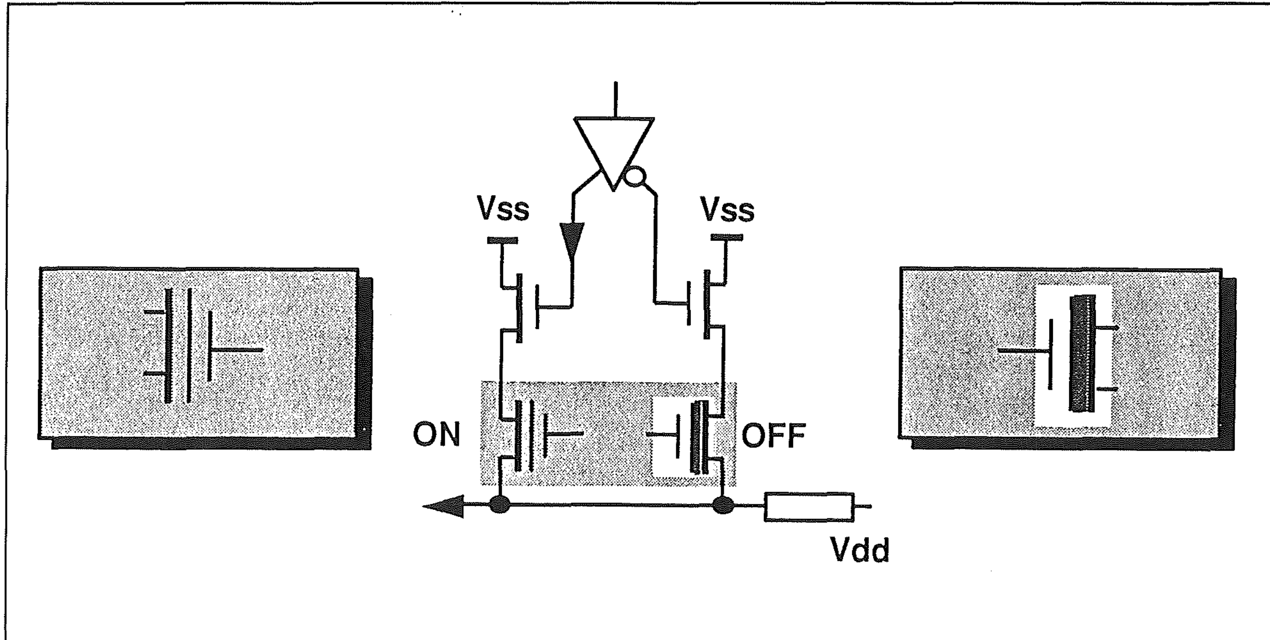


Figuur 3/6.19-30: Een NiCr-zekering, links in ongeprogrammeerde toestand, recht in geprogrammeerde, onderbroken toestand.



Figuur 3/6.19-31: Een zekering volgens de AIM-technologie.

8.9 Software voor de ontwerper

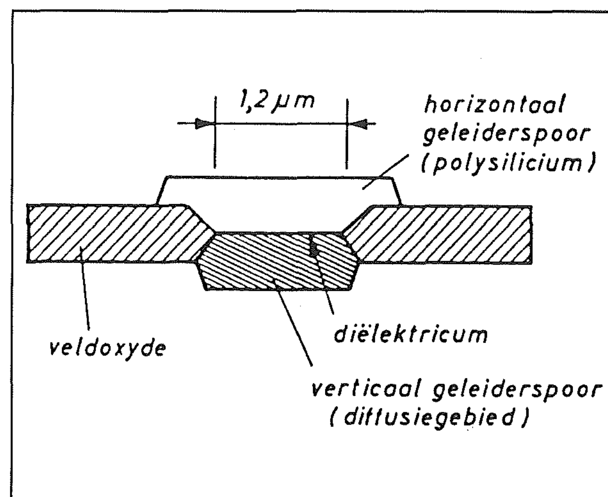


Figuur 3/6.19-32: Een zekering in CMOS-technologie.

De zekering bestaat uit een speciale CMOS-transistor, die in niet-geprogrammeerde toestand niet geleidend is. Bij het programmeren wordt de gate geladen met zogenoemde "hete elektronen". Daardoor gaat de transistor geleiden en de verbinding tussen een horizontale en een vertikale lijn komt tot stand. De hete elektronen blijven in de junctie opgesloten, maar kunnen verwijderd worden door het belichten van de chip met UV-licht. Ook hier programmeert men dus verbindingen en geen onderbrekingen!

De PLICE-zekering

PLICE is het letterwoord van "Programmable Low Impedance Circuit Element". Deze technologie werd ontwikkeld door Actel Semiconductor en heeft als voordeel dat de plaats die in beslag wordt genomen minimaal is.
De breedte van de PLICE-zekering is niet veel groter dan de afstand tussen de sporen in de gebruikte technologie, dus meestal ongeveer 2 á 1,2 µm. PLICE-zekeringen worden dan ook vooral gebruikt bij PLD's die in LSI (Large Scale Integration) zijn uitgevoerd.



Figuur 3/6.19-33: Een doorsnede door een PLICE-zekering.

De samenstelling van een PLICE-zekering is getekend in figuur 3/6.19-33. De zekering wordt gevormd door een uiterst dun

6.19 Werking en principes van PLD's

laagje diëlectricum tussen het horizontale en vertikale spoor van de matrix. Na de fabricage heeft dit laagje een zeer hoge weerstand, zodat er tussen beide lijnen een impedantie van ongeveer 100 MΩ staat. Het diëlectricum wordt vernietigd door tussen beide lijnen een kortstondige (10 ms) programmeerspanning te zetten met een waarde van 18 V, bij een stroom van 15 mA. Nadien bestaat er tussen de twee lijnen een weerstand van slechts 500 Ω, hetgeen men in de hedendaagse hoogimpedante technologieën kan opvatten als een kortsluiting. Het zal wel zonder nadere toelichting duidelijk zijn dat PLICE-zekeringen eenmalig te programmeren zijn.
Ook bij deze techniek programmeert men dus verbindingen en geen onderbrekingen!

Het programmeren van PLD's

Inleiding
Bij de meeste PLD's wordt geprogrammeerd met een spanning van 11,5 V (+/- 0,5 V). De impuls heeft een breedte van 10 tot 50 μs.
De PLD wordt geprogrammeerd door op bepaalde in- en uitgangen "L" en "H" signalen te zetten. Hierdoor wordt een bepaalde rij en kolom van de matrix geselecteerd. Op een andere pen wordt dan de programmeerpuls gezet. Hoe dat in zijn werk gaat is volledig afhankelijk van het type en dus alleen uit de data-sheet van het betreffende IC af te leiden.
De meeste GAL's hebben een EDIT-pen. Door op deze pen een spanning van ongeveer 15 V te zetten komt het IC in de EDIT-modus en kan men de chip wissen of programmeren.

Invoeren van de gegevens
Voor alle PLD-families wordt tegenwoordig speciale software geleverd, die onder MS-DOS of Windows draait op een PC en waarmee tamelijk automatisch de voorbereidingen voor het programmeren kunnen worden getroffen. Nadien kan men de PLD's programmeren door de chip in een PLD-programmer te stoppen die via een interface verbonden is met de PC.
Afhankelijk van de software moet men de te programmeren schakeling ingeven onder de vorm van:
- Booleaanse vergelijkingen, hetgeen voor eenvoudige zuiver combinatorische schakelingen (alleen product- en som-termen) aan te bevelen is;
- waarheidstabellen, hetgeen voor de programmeur in bepaalde omstandigheden minder werk met zich met mee brengt;
- sequentiële vergelijkingen, die aan te bevelen zijn als men met sequentiële ontwerpen werkt, dus ontwerpen waar flip-flop's in zitten;
- grafische ingave onder de vorm van pulsdiagrammen van alle signalen.

Bekende programmeer software
De bekendste software pakketten die ontwikkeld zijn voor het programmeren van PLD's zijn:
- ABEL:
  wordt op de markt gebracht door Data I/O en ondersteunt PLD's van verschillende fabrikanten;
- AMAZE:
  is een pakket van Philips, waarmee men alle PLD's van deze fabrikant (Signetics) kan programmeren;
- LOG/IC:

wordt geleverd met een uitgebreide bibliotheek, die naarmate er meer PLD's op de markt komen wordt uitgebreid;
- PALASM:
  een pakket van AMD, waarmee echter alleen de eigen PLD's van deze fabrikant te programmeren zijn;
- SNAP:
  een pakket van Philips waarmee alleen eigen PLD's te programmeren zijn en dat voorzien is van uitgebreide fout-simulatoren.

De Logic Definition Listing
Alle pakketten genereren een zogenoemde "Logic Definition Listing". In deze lijst staan alle relevante gegevens, die het programmeerapparaat nodig heeft om een PLD te kunnen programmeren. Een voorbeeld van een dergelijke LDL is gegeven in figuur 3/6.19-34.

Na wat algemene gegevens en regels met commentaar volgt de PINLIST. Hierin worden alle pennen van de schakeling volledig gedefinieerd, zodat het programmeerapparaat weet welke signalen aan welke pennen aangelegd moet worden.
Nadien volgt de LOGIC EQUATION, de lijst waarin de logische Booleaanse formules voor iedere uitgang worden beschreven.
Hiervoor worden gestandaardiseerde coderingen gebruikt:
- ;
  alles na de punt/komma is kommentaar;
- /
  invertering;
- *
  AND-knoop;
- +
  OR-knoop;
- :+:
  EXOR-knoop;
- ()
  voorwaardelijke tri-state;
- =
  toekenning aan uitgang;
- :=
  toekenning aan uitgang bij de volgende clock-puls.

JEDEC-file
Uit de Logic Definition Listing genereert de software de zogenoemde JEDEC-file, die via een seriële poort naar het programmeerapparaat wordt gestuurd.
Deze JEDEC-file is niet erg inzichtelijk, zoals blijkt uit het voorbeeld van figuur 3/6.19-35.

Het JEDEC-formaat werd oorspronkelijk ontwikkeld door DATA I/O voor de eigen PLD-programmers. Inmiddels is het een de factor standaard geworden.
Alle JEDEC-kommando's worden ingeleid met een *, nadien volgen een of twee letters, die de volgende betekenis hebben:
- N
  kommentaar;
- QF
  totaal aantal zekeringen in het IC;
- QP
  aantal pennen van het IC:
- QV
  maximaal aantal testvectoren;
- F
  default toestand van een zekering;
- C
  genereren van een positieve flank;
- K
  genereren van een negatieve flank;
- N
  niet te testen pennen, zoals massa en voeding;
- H
  hoog logisch niveau;

6.19 Werking en principes van PLD's

```
@DEVICE TYPE
PLHS16L8
@DRAWING
@REVISION
       01
@DESCRIPTION
       BCD-SEVEN-SEGMENT-DECODER WITH DOT
       (HEXADECIMAL DISPLAY)

@PINLIST

"<--------FUNCTION-------->   <--REFERENCE-->"
"PINLABEL     PIN #  PIN_FCT   PIN_ID     OE_CTRL"
D             1        I        I0         -  ;
C             2        I        I1         -  ;
B             3        I        I2         -  ;
A             4        I        I3         -  ;
N/C           5        I        I4         -  ;
N/C           6        I        I5         -  ;
N/C           7        I        I6         -  ;
N/C           8        I        I7         -  ;
N/C           9        I        I8         -  ;
GND          10        0V       GND        -  ;
N/C          11        I        I9         -  ;
DOT          12        /O       OA         EA ;
OG           13        /O       B0         D0 ;
OF           14        /O       B1         D1 ;
OE           15        /O       B2         D2 ;
OD           16        /O       B3         D3 ;
OC           17        /O       B4         D4 ;
OB           18        /O       B5         D5 ;
OA           19        /O       OB         EB ;
VCC          20        +5V      VCC        -  ;

@LOGIC EQUATION

OA  = /( A* C*/D + /A*/C  + /A* D   +  B* C    +  B*/D + /B*/C*D) ;
OB  = /( A* B*/D +  A*/B*D + /A*/B*/D + /A*/C    + /B*/C ) ;
OC  = /( A*/B    +  A*/C  + /B*/D   + /C* D    +  C*/D ) ;
OD  = /( A* B*/C +  A*/B*C + /A* B* C + /A*/C*/D + /B* D ) ;
OE  = /(/A* B    + /A*/C  +  B* D   +  C* D    ) ;
OF  = /(/A*/B    + /A* C  +  B* D   + /B* C*/D + /C* D ) ;
OG  = /( A* D    + /A* B  +  B*/C   + /B* C*/D + /C* D ) ;
DOT = /( B* D    +  C* D    ) ;
```

Figuur 3/6.19-34: Een voorbeeld van een Logic Definition Listing.

8.9 Software voor de ontwerper

```
PAL20L8
ATARI LINK ADAPTER (ALIA)
*QP24
*QF3200
<STX>
*G0*F0
*L0000 1111 1111 1111 1111 1111 1111 1111 1111 1111 1111
*L0040 1111 1111 0111 1111 1111 1111 1111 0111 0111 1111
*L0640 1111 1111 1111 1111 1111 1111 1111 1111 1111 1111
*L0680 1111 1111 1111 1111 1111 1111 1111 1111 1111 1011
*L0720 1111 1111 1111 1111 1111 1111 1111 1111 1111 1110
*L0960 1111 1111 1111 1111 1111 1111 1111 1111 1111 1111
*L1000 1110 1111 1111 1111 1111 1111 1111 0111 0111 1011
*L1040 1110 1111 1111 1111 1111 1111 1111 0111 0111 1110
*L1280 1111 1111 1111 1111 1111 1111 1111 1111 1111 1111
*L1320 1011 0110 1111 1111 1111 1111 1011 1111 1111 1111
*L1360 1011 1001 1111 1111 1111 1111 1111 1111 1110 1111
*L1400 1011 0101 1111 1111 1111 1011 1111 1111 1111 1111
*L1440 0111 1110 1111 1111 1111 1011 1111 1111 1111 1111
*L1480 0111 1101 1111 1111 0111 1111 1111 1111 1111 1111
*L1520 1101 1111 1111 1111 1111 1011 1111 0111 1111 1111
*L1600 1111 1111 1111 1111 1111 1111 1111 1111 1111 1111
*L1640 1101 1111 1111 1111 1111 1111 1111 0111 1111 1011
*L1680 1101 1111 1111 1111 1111 1111 1111 0111 1111 1110
*L1920 1111 1111 1111 1111 1111 1111 1111 1111 1111 1111
*L1960 1111 1111 1111 0111 1111 1111 1111 1111 1101 1111
*L2000 1101 1111 1111 0111 1111 1111 1111 1111 1110 1111
*L2040 1110 1111 1111 1011 1111 1111 1111 1111 1101 1111
*L2080 1110 1111 1111 1111 0111 1111 1111 1111 1111 1111
*L2120 1010 1111 1111 1111 1111 1111 1111 1111 1111 1111
*L2160 1111 1111 1111 1111 1111 1111 1111 1011 1111 1111
*L2240 1111 1111 1111 1111 1111 1111 1111 1111 1111 1111
*L2280 1111 1111 1111 1111 1111 1101 1111 1111 1111 1111
*C80D7
*<ETX>1A12
```

Figuur 3/6.19-35: De zogenoemde JEDEC-file wordt via de seriële poort van de computer naar de PLD-programmer gestuurd.

6.19 Werking en principes van PLD's

- L
  laag logisch niveau
- X
  logisch niveau is verder onbelangrijk;
- Z
  testen op hoog-ohmige toestand van een uitgang.

Nadien volgen de programmeergegevens voor de zekeringen.
Deze data zijn opgenomen tussen twee regels met de coderingen <STX> (start) en <ETX> (einde).
De programmeergegevens staan in lange regels, waarbij een 1 staat voor een te programmeren zekering en een 0 voor een niet te programmeren zekering. Deze regels worden voorafgegaan door de basis-code van de zekeringen die in de regel geprogrammeerd worden. Dit getal wordt voorafgegaan door de codering *L.

Een voorbeeld

HEX-display

Tot slot van deze bespreking wordt een eenvoudig programmeringsvoorbeeld besproken. Zoals men weet kan men met vier logische signalen A, B, C en D in totaal 16 logische combinaties samenstellen. Deze worden in de hexadecimale notatie voorgesteld door de cijfers 0 tot en met 9 en door de letters A tot en met F. Hoewel er tal van IC's zijn die de BCD-code omzetten naar besturingssignalen voor zeven-segment display's is dat, vreemd genoeg, niet het geval voor de HEX-code. Maar dank zij PLD's hoeft men niet te treuren, want een dergelijke klus is als het ware geschapen voor een eenvoudige PLD, zoals een AND/OR-PAL van de eerste generatie.

Schema

Het schema van het HEX-display is getekend in figuur 3/6.19-36.

De vier HEX-signalen A, B, C en D worden voorgesteld door vier schakelaartjes, die omschakelen tussen "L" en "H". Deze signalen gaan rechtstreeks naar de ingangen van de PLD. De acht uitgangen sturen via een buffer 74LS245 en serieweerstanden de LED's in het zeven-segment display met gemeenschappelijke anode. De segmenten gaan dus branden als de uitgangen OA tot en met DOT naar "L" worden getrokken.

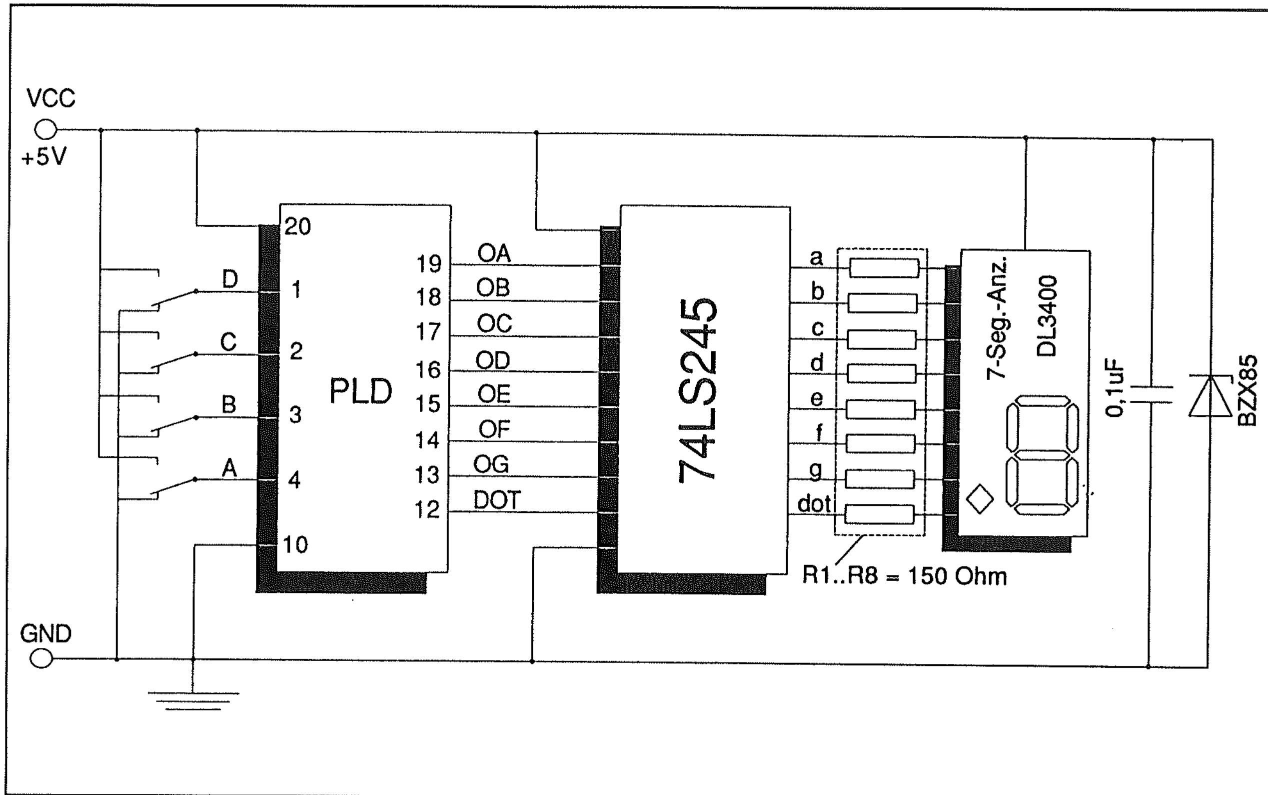
De waarheidstabel

Allereerst wordt de waarheidstabel van de schakeling opgesteld. In deze tabel, zie figuur 3/6.19-37, worden de 16 ingangscondities van de signalen A, B, C en D vastgelegd en wordt genoteerd welke segmenten voor iedere conditie moeten branden.

Selectie PLD

Uit de waarheidstabel blijkt dat deze schakeling zuiver combinatorisch is. Voor iedere combinatie van ingangssignalen staat slechts één combinatie van uitgangssignalen. Een dergelijk probleem kan volledig opgelost worden door gebruik te maken van product- en som-termen, dus van AND - en OR-poorten. Voor de PLD kan men dus een AND/OR-PAL van de allereerste generatie gebruiken. Het zal verder duidelijk zijn dat deze PAL minstens vier ingangen en acht uitgangen moet hebben. Een geschikt exemplaar is de 16L8, men zou bijvoorbeeld het type PLHS16L8 van Signetics kunnen toepassen. De aansluitgegevens en het interne blokschema staan vermeld in figuur 3/6.19-38.

8.9 Software voor de ontwerper



Figuur 3/6.19-36: Het schema van het HEX-display met een PLD als decoder.

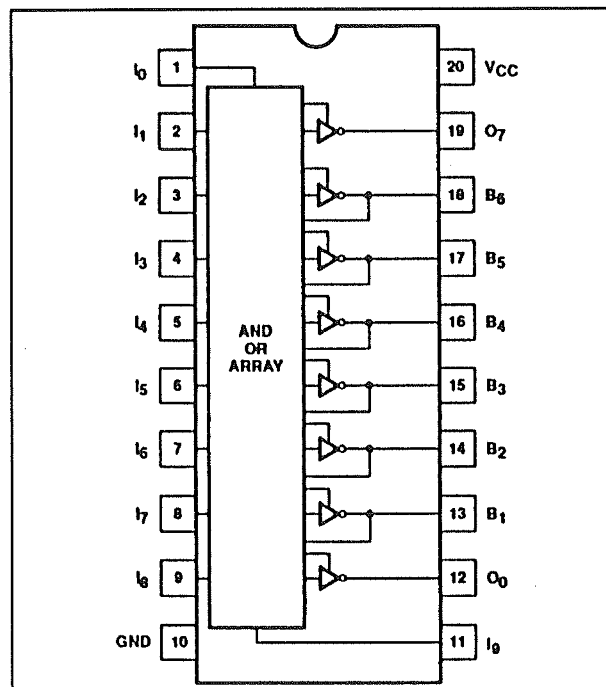
LABEL PIN / CHAR	D 1	C 2	B 3	A 4	OA 19	OB 18	OC 17	OD 16	OE 15	OF 14	OG 13	DOT 12
0	0	0	0	0	L	L	L	L	L	L	H	H
1	0	0	0	1	H	L	L	H	H	H	H	H
2	0	0	1	0	L	L	H	L	L	H	L	H
3	0	0	1	1	L	L	L	L	H	H	L	H
4	0	1	0	0	H	L	L	H	H	L	L	H
5	0	1	0	1	L	H	L	L	H	L	L	H
6	0	1	1	0	L	H	L	L	L	L	L	H
7	0	1	1	1	L	L	L	H	H	H	H	H
8	1	0	0	0	L	L	L	L	L	L	L	H
9	1	0	0	1	L	L	L	L	H	L	L	H
.A	1	0	1	0	L	L	L	H	L	L	L	L
.B	1	0	1	1	H	H	L	L	L	L	L	L
.C	1	1	0	0	L	H	H	L	L	L	H	L
.D	1	1	0	1	H	L	L	L	L	H	L	L
.E	1	1	1	0	L	H	H	L	L	L	L	L
.F	1	1	1	!	L	H	H	H	L	L	L	L

a
f b
g
e c
DOT d

ACTIV LOW
(common Anode)

Figuur 3/6.19-37: De waarheidstabel van de schakeling.

6.19 Werking en principes van PLD's



Figuur 3/6.19-38: De aansluitgegevens van de PLHS16L8 van Signetics (Philips).

Het opstellen van de Logic Equation

Voor het besturen van de software en de PLD-programmer heeft men de Logic Equation van de schakeling nodig. Deze is gegeven in figuur 3/6.19-39.

In deze lijst wordt de Booleaanse uitdrukking opgeschreven voor iedere PAL-uitgang. Wie goed thuis is in de Booleaanse algebra zal weinig moeite hebben met het oplossen van dit eenvoudige voorbeeld. Maar ook zonder kennis van deze algebra kan men, door wat te puzzelen, toch deze uitdrukkingen samenstellen. Als voorbeeld wordt de eenvoudigste uitdrukking behandeld, deze voor de decimale punt DOT. Deze punt gaat branden als het display de letters van de HEX-code weergeeft. Men moet dus in de waarheidstabel de regels bekijken van .A, .B, .C, .D, .E en .F. Dat zijn uiteraard de enige regels, waarvoor DOT "L" moet worden. In iedere regel moet men nu een unieke combinatie van logische niveaus van A, B, C en D opsporen. Met uniek wordt bedoeld dat deze combinatie nergens mag voorkomen waar DOT niet "L" is. Vaak zal het noodzakelijk zijn ook de geïnverteerde ingangs-operatoren bij deze combinatie te betrekken.

- De regel .A:
  Hier is duidelijk dat de combinatie D = "H" en B = "H" bruikbaar is. Deze combinatie komt namelijk niet voor in de regels 0 tot en met 9.
- De regel .B:
  Hier komt de genoemde combinatie wél voor, maar dat is niet erg want ook nu moet DOT naar "L".
- De regel .C:
  Hier is de combinatie D = "H" en C = "H" uniek en deze wordt dan ook gebruikt voor het sturen van de DOT.
- De regel .D:
  Men kan dezelfde combinatie gebruiken als voor .C.
- De regel .E:
  Weer is de combinatie D = "H" en B = "H" te gebruiken.
- De regel .F:
  Idem.

Besluitend kan men stellen dat men het laag worden van de uitgang DOT als volgt kan samenvatten: ***de uitgang DOT wordt laag als D = "H" en B = "H" of als D = "H" en C = "H"***. Dit is dus een schoolvoorbeeld van twee product-termen die tot een somterm verenigd worden, de basis van PAL-programmering!

In Booleaanse notatie:

DOT = [B * D] + [D * C]

Dit is precies dezelfde uitdrukking die voor DOT terug te vinden is in de Logic Equation van figuur 3/6.19-39, maar dan geschreven volgens de gestandaardiseerde methode die voor de Logic Equation gebruikt wordt.

8.9 Software voor de ontwerper

```
@LOGIC EQUATION

OA  = /( A* C*/D + /A*/C  + /A* D    +  B* C    +  B*/D + /B*/C*D) ;
OB  = /( A* B*/D +  A*/B*D + /A*/B*/D + /A*/C    + /B*/C ) ;
OC  = /( A*/B    +  A*/C  + /B*/D    + /C* D    +  C*/D ) ;
OD  = /( A* B*/C +  A*/B*C + /A* B* C + /A*/C*/D + /B* D ) ;
OE  = /(/A* B    + /A*/C  +  B* D    +  C* D    ) ;
OF  = /(/A*/B    + /A* C  +  B* D    + /B* C*/D + /C* D ) ;
OG  = /( A* D    + /A* B  +  B*/C    + /B* C*/D + /C* D ) ;
DOT = /( B* D    +  C* D  ) ;
```

Figuur 3/6.19-39: De Logic Equation van het HEX-display.

Op dezelfde manier zou men de formules voor de zeven andere uitgangen kunnen opstellen, al zijn deze uiteraard wel wat ingewikkelder.

Maar voor de echte logische puzzelaar is dit dé uitdaging van de digitale elektronica!

3/6.20

Werking en principes van transceivers

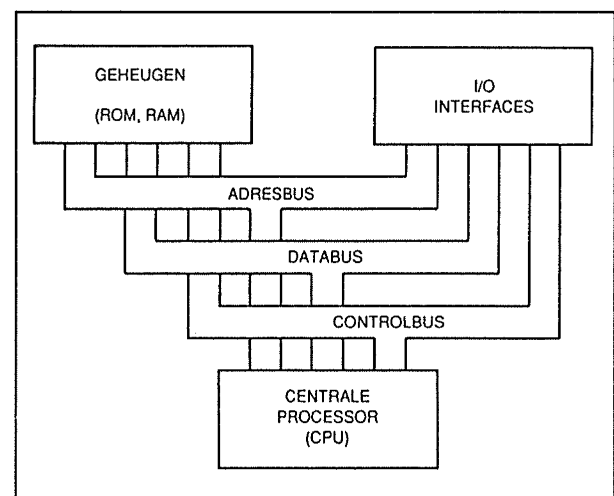
Werking en principes

Bussen

In de moderne digitale techniek wordt veelvuldig gebruik gemaakt van zogenaamde "bussen". Dit zijn structuren waarbij meerdere parallelle lijnen dienen om gegevens, besturingssignalen en adressen te transporteren, bijvoorbeeld tussen een microprocessor en het geheugen, tussen in- en uitgangsschakelingen, registers, enzovoorts. Meestal worden 4, 8, 16, 32 of 64 parallelle lijnen gebruikt, waarbij men dan bijvoorbeeld spreekt van een "8 bit brede bus".
Er wordt bij (micro)computers onderscheid gemaakt tussen de adresbus, databus en controlbus, zodat het volledige bussysteem kan worden voorgesteld als geschetst in figuur 3/6.20-1.

Datarichting op de bussen

De informatie op de adresbus, de adressen die naar andere blokken gestuurd moeten worden, is altijd afkomstig van de processor en wordt zodoende unidirectioneel (één-richting) genoemd. De informatie op de controlbus dient voor de zogenoemde "timing en control".
Bij kleine computers worden deze signalen alleen door de processor geleverd, maar bij grotere vindt transport in twee richtingen plaats, hetgeen bidirectioneel wordt genoemd. De databus is altijd bidirectioneel, aangezien gegevens zowel van als naar de processor getransporteerd moeten worden. Het is hierbij noodzakelijk dat de data door slechts één schakeling tegelijk op de bus wordt gezet.

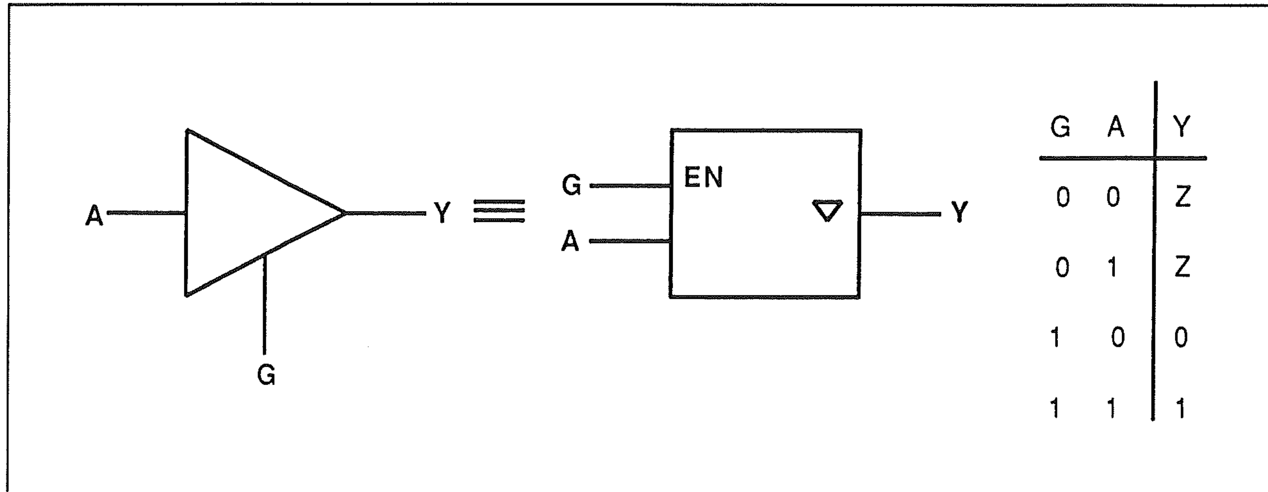


Figuur 3/6.20-1: Een eenvoudig computermodel met de bidirectionele databus en controlbus en de unidirectionele adresbus.

Besturing van de bus

De lijnen van de bus zijn aangesloten op verschillende schakelingen. Het kan dus zijn dat één lijn door drie of vier schakelingen belast wordt. Nu zal het duidelijk zijn dat het verboden is dat uitgangen van verschillende schakelingen zonder meer parallel aan elkaar worden geschakeld door middel van de bus-lijn.

6.20 Werking en principes van transceivers



G	A	Y
0	0	Z
0	1	Z
1	0	0
1	1	1

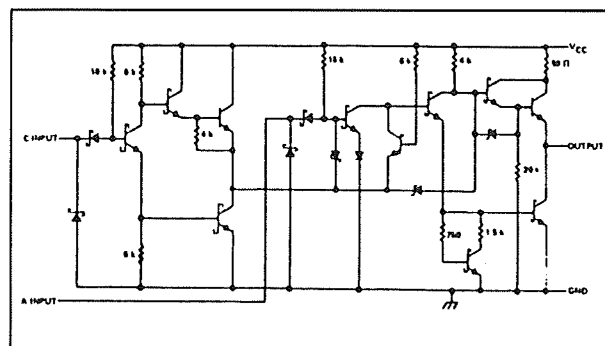
Figuur 3/6.20-2: Het oude en nieuwe logisch symbool en de waarheidstabel van een tri-state buffer.

Er moeten dus speciale schakelingen tussen de lijnen van de bussen en de op de bus aan te sluiten schakelingen worden gezet, die alleen die in- en uitgangen met de bus-lijn verbinden, die op een bepaald moment via de bus gegevens moeten uitwisselen. Bovendien moet men steeds de richting van de gegevens door de bus-lijn instellen. Als de bus bijvoorbeeld wordt gebruikt om gegevens tussen de microprocessor en het geheugen uit te wisselen, dan zal de ene keer het geheugen de "ingang" van de bus zijn en de processor de "uitgang". In een ander geval zal echter de processor de "ingang" zijn en het geheugen de "uitgang" van de bus.
Om dit soort problemen zo eenvoudig mogelijk op te lossen heeft men speciale schakelingen ontwikkeld, de zogenoemde "transceivers". Deze worden tussen de lijnen van de bus en de bus-belastingen geschakeld en zorgen ervoor dat schakelingen die niet aan het bus-verkeer deelnemen van de bus worden los gekoppeld. Daarnaast bepalen de transceivers welke schakelingen gegevens op de bus zetten (ingang zijn) en welke schakelingen de gegevens van de bus halen (uitgang zijn).

Naast de noodzakelijke in- en uitgangen hebben transceivers dus steeds twee besturingsingangen, die respectievelijk de richting van het verkeer bepalen (DIR) en de transceiver in- of uitschakelen (GRANT).

De tri-state buffer als basis

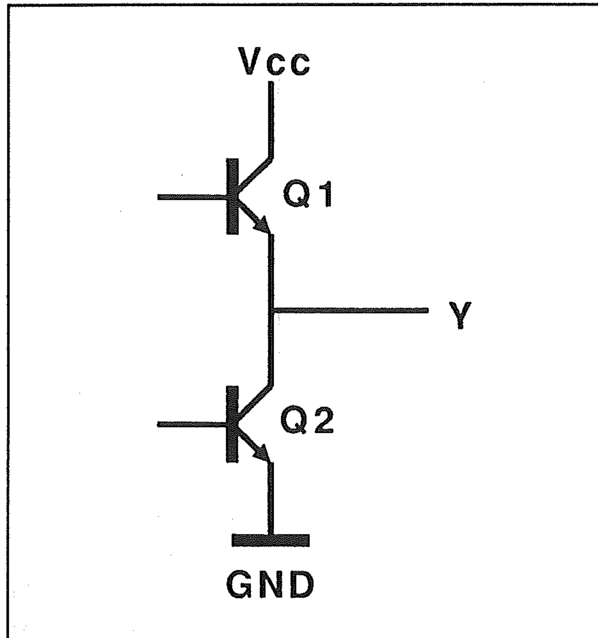
De schakeling waarvan in figuur 3/6.20-2 het logisch symbool en de waarheidstabel zijn getekend, maakt het mogelijk om data uit verschillende bronnen op dezelfde bus te zetten.



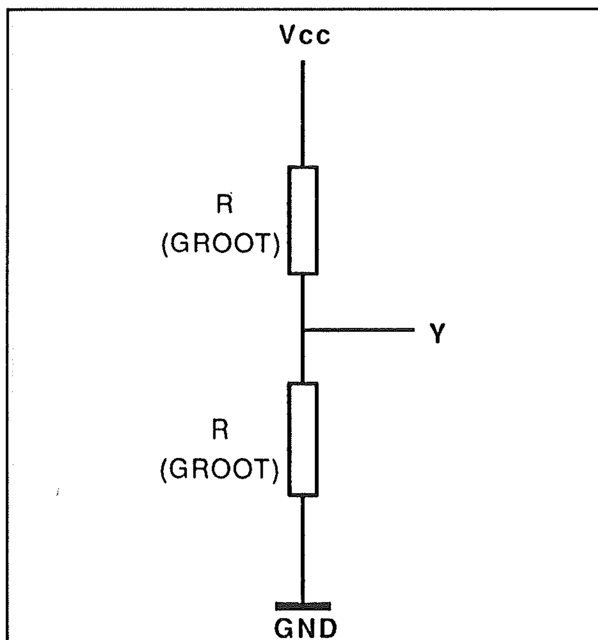
Figuur 3/6.20-3: Het inwendige uitgewerkte schema van een lijn van een busbuffer met tri-state uitgang.

Het is een buffer met een zogenaamde "tri-state"-uitgang.

6.20 Werking en principes van transceivers



Figuur 3/6.20-4: Vereenvoudigde voorstelling van de laatste trap van een tri-state buffer.



Figuur 3/6.20-5: Het vervangingsschema van de laatste trap van een tri-state buffer in sperrende toestand.

Wanneer de Enable-ingang (ook Grant- of C-ingang genoemd) HOOG is, is de uitgang afhankelijk van de data-ingang HOOG of LAAG. Is de Enable-ingang LAAG, dan is de uitgang ***altijd*** hoog-impedant of zwevend. In figuur 3/6.20-3 is getekend hoe zo'n schakeling in de praktijk wordt opgebouwd, terwijl figuur 3/6.20-4 alleen de laatste trap ervan weergeeft.

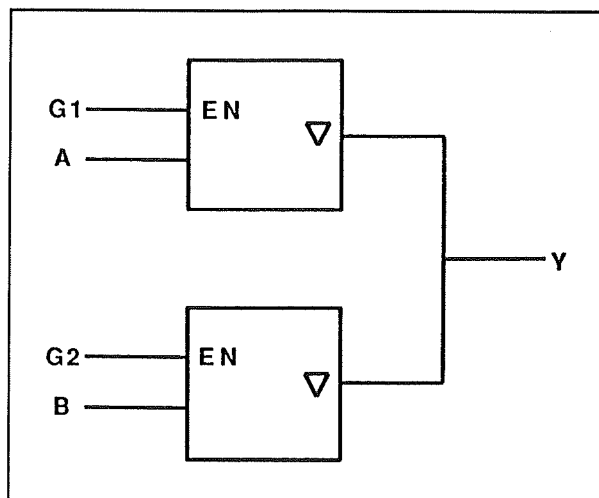
Wanneer transistor Q1 geleidt is uitgang Y "H", geleidt transistor Q2, dan is de uitgang "L". Geleiden van Q1 en Q2 tegelijkertijd zou een kortsluiting van V_{CC} naar GND betekenen en mag dus nooit voorkomen. Wel kunnen Q1 en Q2 tegelijk sperren, waardoor de uitgang van de schakeling naar de derde mogelijkheid, namelijk 3- of tri-state schakelt. In dat geval zal de uitgang geen stroom meer kunnen leveren of opnemen zodat gezegd wordt dat de uitgang "zwevend" is. Figuur 3/6.20-5 geeft het vervangingsschema van de uitgang voor de tri-state toestand. Men kan de uitgang dan vervangen door niets meer of minder dan twee zeer hoge weerstanden, geschakeld tussen de uitgang en de massa en de voedingsspanning. De uitgang Y staat weliswaar op een bepaalde willekeurige spanning ergens tussen de voeding en de massa, maar schakelt men de uitgang op een per definitie laagimpedante bus, dan zal deze spanning onmiddellijk wegvallen vanwege de zeer hoge waarde van de weerstanden. Door de uitgangstrap loopt slechts een zeer kleine lekstroom en de uitgang is "hoog-impedant".

Constructie van de bus

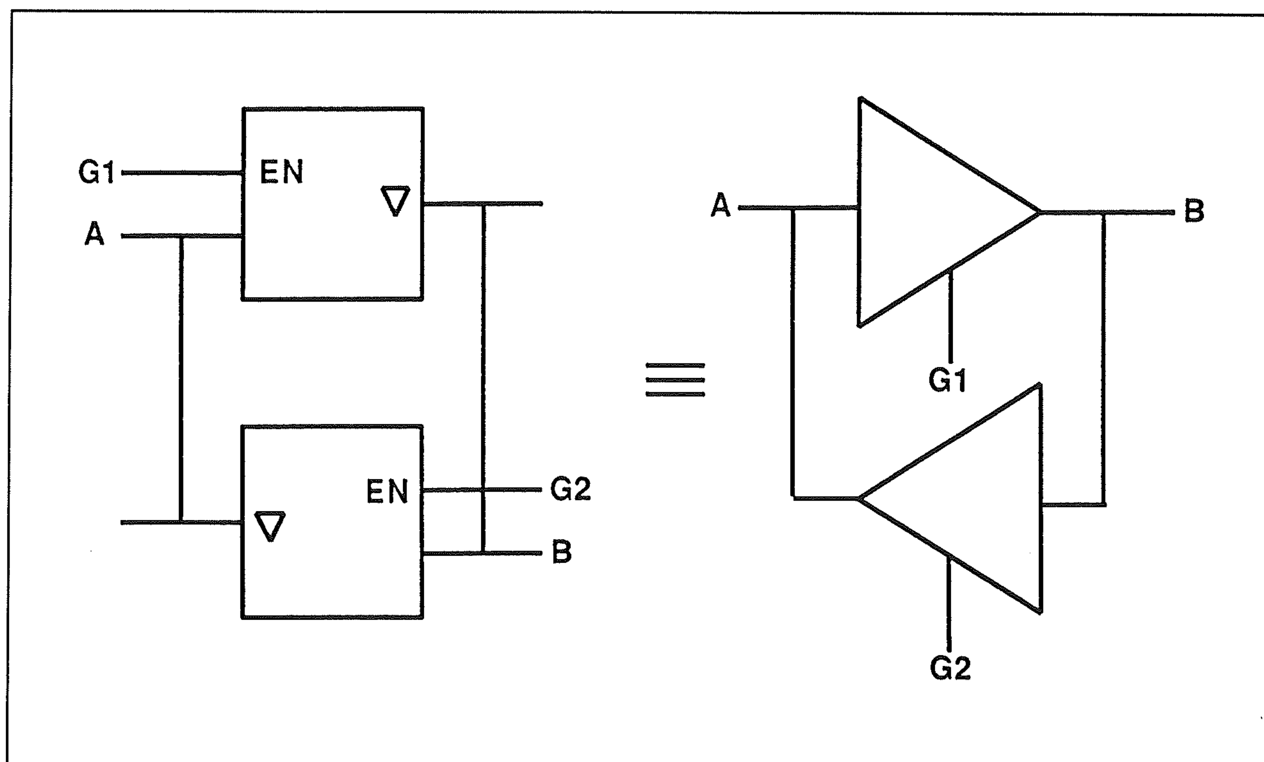
Voor het construeren van een bus-lijn worden meerdere tri-state schakelingen gebruikt, zoals getekend in figuur

6.20 Werking en principes van transceivers

3/6.20-6. De uitgangen kunnen nu met elkaar worden verbonden als slechts één schakeling tegelijk "meester" of "master" van de bus is. Als bijvoorbeeld G1 HOOG is, dan komt de informatie van ingang A op de bus-lijn Y. In dat geval moet G2 LAAG zijn om de ingang B te sperren. Hierna kan bijvoorbeeld ingang B naar de bus-lijn worden geschakeld, waarbij ingang A uiteraard gesperd moet worden. Het is mogelijk om nog meer schakelingen op dezelfde Y-lijn aan te sluiten, als er maar op gelet wordt dat er op een bepaald moment slechts één meester kan zijn.

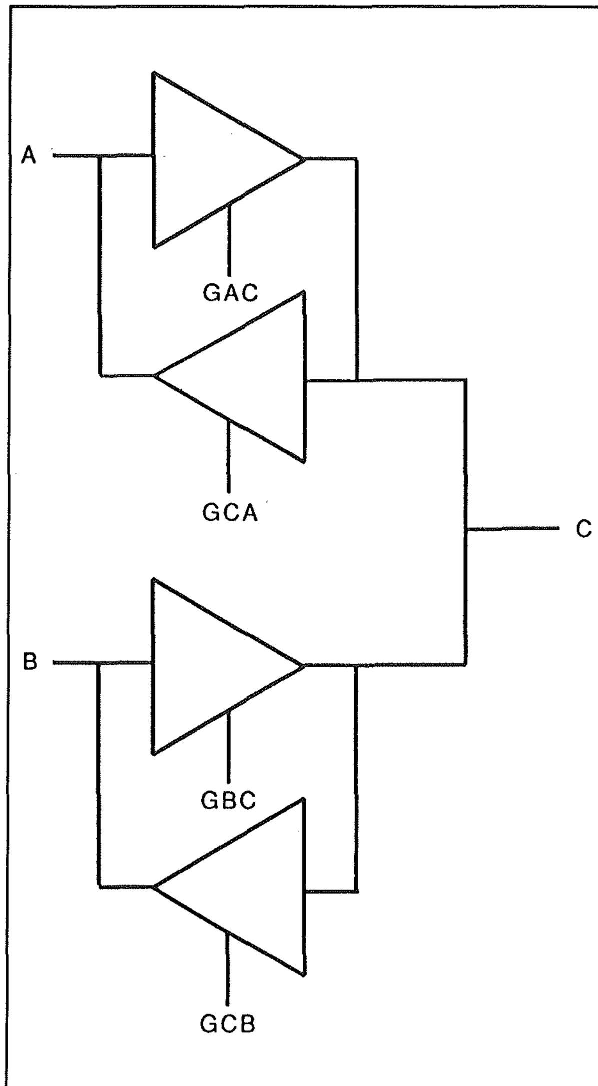


Figuur 3/6.20-6: De schematische samenstelling van één data-lijn van een unidirectionele bus, waarmee data van A naar Y of van B naar Y (bij G1 respectievelijk G2 "L") gestuurd kan worden.

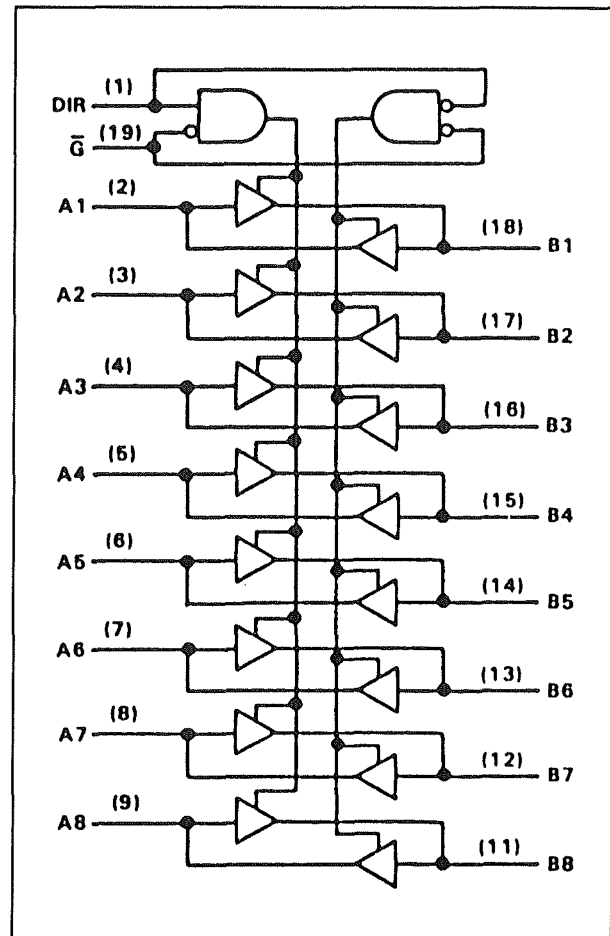


Figuur 3/6.20-7: De fundamentele samenstelling en werking van een transceiver. Data gaat van A naar B als G1 "H" is en van B naar A als G2 "H" is.

6.20 Werking en principes van transceivers



Figuur 3/6.20-8: Twee transceivers zijn noodzakelijk om een bus-lijn C met twee schakelingen A en B te verbinden.



Figuur 3/6.20-9: Het intern blokschema van de 8 bit transceiver 74HC245.

Transceivers

Tot nu toe ging de data slechts één kant op en was er sprake van zogenaamde "busbuffers" of "busdrivers". Met dezelfde onderdelen is het ook mogelijk om de data naar keuze van links naar rechts of andersom te laten gaan.
Met draait daartoe, zoals getekend in figuur 3/6.20-7, één van de tri-state buffers om.

Is G1 nu "H" en G2 "L", dan gaat de data van A naar B en is G1 "L" en G2 "H", dat gaan de gegevens van B naar A. Zijn G1 en G2 beide "L", dan zijn A en B volledig van elkaar gescheiden en staat de schakeling dus in tri-state.
Op deze wijze is een "transceiver" ontstaan, een woord dat ontstaat als men het begin en het einde van de woorden ***TRANS***mitter en re***CEIVER*** samen trekt.

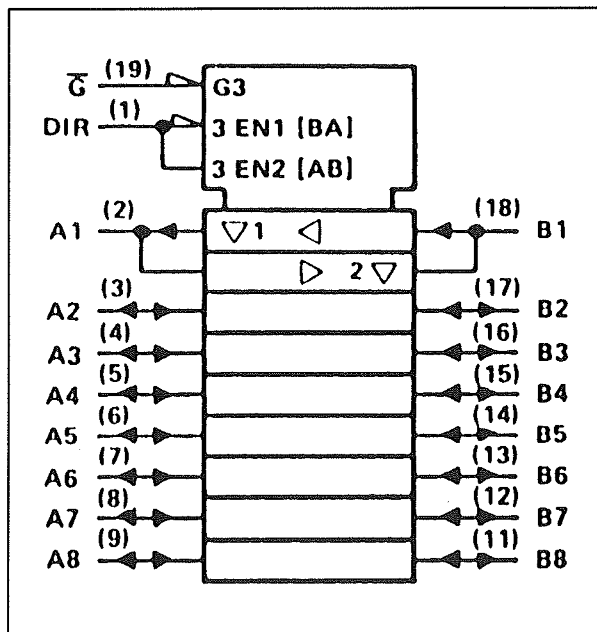
Transceivers in de praktijk

In figuur 3/6.20-8 is een uitbreiding getekend, waarbij twee transceivers worden gebruikt om een bus-lijn C met twee schakelingen A en B te verbinden.

6.20 Werking en principes van transceivers

CONTROL INPUTS		OPERATION
$\overline{G}$	DIR	
L	L	B data to A bus
L	H	A data to B bus
H	X	Isolation

Figuur 3/6.20-10: De waarheidstabel van de 74HC245.



Figuur 3/6.20-11: Het logisch symbool van een transceiver, in dit voorbeeld van de 74HC245.

Door de werkingsrichting van de transceivers te besturen kan men met dit systeem gegevens via de bus-lijn van A naar B sturen of van B naar A. Bovendien kan men natuurlijk de transceivers ook in tri-state sturen, waardoor zowel A als B losgekoppeld worden van de bus-lijn en andere schakelingen toegang tot de bus-lijn kunnen krijgen. Het getekende systeem wordt steeds in de praktijk toegepast om bussen met schakelingen te verbinden.

Voorbeeld

De 74HC245
Als voorbeeld van de praktische opbouw van een transceiver wordt de 74HC245 besproken.

Intern blokschema
Het intern blokschema van dit IC is getekend in figuur 3/6.20-9.
Hieruit blijkt duidelijk dat het IC uit niets anders bestaat dan acht basisschakelingen volgens figuur 3/6.20-7. De ingangen worden A genoemd, de uitgangen B. Daarnaast beschikt het IC over twee besturingsingangen DIR en $\overline{G}$.

De waarheidstabel
De waarheidstabel van de 74HC245 is getekend in figuur 3/6.20-10. Hieruit blijkt dat een "L" op $\overline{G}$ de schakeling "transparant" maakt. De schakeling kan dan werken en de A's met de B's doorverbinden. Is $\overline{G}$ "H", dan is de schakeling "niet-transparant" en staan alle A's en alle B's in tri-state.
De richting van de transparante werking wordt bepaald door het signaal DIR. Is dit "L", dan gaan de gegevens van B naar A. Is dit besturingssignaal "H", dan gaan de gegevens van A naar B. Aan de A-zijde kan zich nu bijvoorbeeld een processor bevinden, met aan de B-zijde een geheugen. De processor kan dan data in het geheugen schrijven en deze later ook weer ophalen.

Logisch symbool van een transceiver
Het logisch symbool van de 74HC245, maar ook van iedere andere transceiver, is getekend in figuur 3/6.20-11.

De bidirectionele werking wordt voorgesteld door de pijltjes in de A- en B-lijnen.

3/6.21

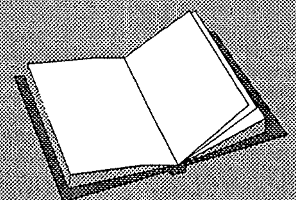
Het transport van digitale signalen

Inleiding

Het probleem van de beïnvloeding
Al tientallen jaren worden de ontwerpers van digitale schakelingen en systemen geconfronteerd met het vraagstuk van het transport van digitale signalen over allerlei afstanden met steeds hogere snelheden.
Op printkaarten vindt signaal-overdracht meestal direkt plaats van de ene logische schakeling naar de andere. Low Power Schottky TTL en HCMOS kunnen bijvoorbeeld werken bij klokfrequenties van maximaal 40 MHz. Op een print bedraagt de te overbruggen afstand van de signalen meestal slechts enkele tientallen centimeter.
Toch moet er ook daar al op worden gelet dat de verbindingen zo kort mogelijk zijn, dat de voeding ontkoppeld is en vooral dat er goede en laag-impedante aardverbindingen zijn. Met "aardverbindingen" wordt eigenlijk de nul van de voeding bedoeld, omdat het in de praktijk gebruikelijk is de hele schakeling tijdens de ontwerpfase te laten zweven, waarna het meest geschikte aardpunt kan worden gekozen. Uiteraard kan dan worden beslist dat eventueel alle nulpunten van alle printkaarten en connectoren met bijvoorbeeld de (geaarde) kast moeten worden verbonden.

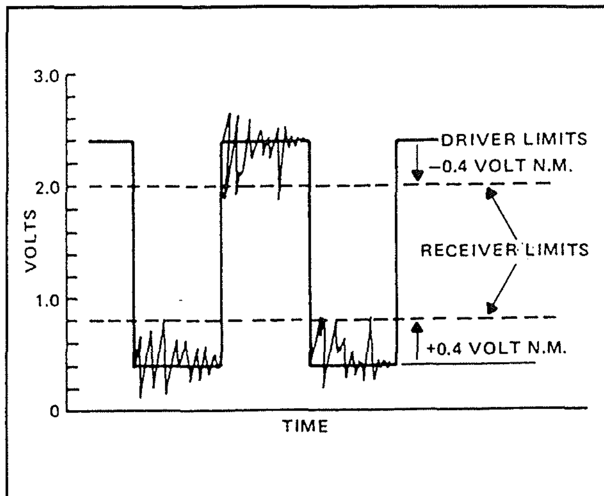
Transport tussen printplaten
Wanneer een (gesloten) apparaat uit meerdere printkaarten of subsystemen is opgebouwd, wordt het data-transport veel kritischer. De afstand die moet worden overbrugd is dan immers groter, waardoor de kans tot oppikken van stoorsignalen ook groter wordt. De gegarandeerde stoordrempel van +/-0,4 V van standaard TTL (figuur 3/6.21-1) is dan vaak onvoldoende, zodat speciale schakelingen, lijndrivers en -receivers genoemd, moeten worden toegepast. Daarnaast kan men de signaallijnen gaan afschermen of "twisten". Dit strak om elkaar heen draaien van aderparen heeft natuurlijk alleen zin als de signalen even groot maar tegengesteld zijn, zodat ze elkaar kunnen opheffen. En in plaats van afschermen wordt vaak een bandkabel gebruikt waarbij signaal en aarde elkaar telkens afwisselen.

LEES OOK:
Hoofdstuk 3/6.16
Hoofdstuk 3/6.20
Hoofdstuk 6/6.11



6.21 Het transport van digitale signalen

Deze methode werd bijvoorbeeld reeds toegepast bij de "Unibus" in de PDP-11 computers van DEC.



Figuur 3/6.21-1: De storingsmarges van standaard TTL-schakelingen.

Van apparaat naar apparaat
De data-overdracht wordt natuurlijk nog kritischer wanneer verschillende apparaten op elkaar worden aangesloten. Een computer heeft bijvoorbeeld een monitor, een toetsenbord en een printer nodig. De signalen staan dan bloot aan storingen van buitenaf en door de grotere lengte van de kabels is de kans tot oppikken nog groter. Dan wordt de toevlucht genomen tot speciale maatregelen: standaard protocollen, zoals RS-232, -422, -423 of -485 (en binnenkort de USB, oftewel de Universele Seriële Bus) en het gebruik van speciale drivers en receivers daarvoor.

Algemene eisen
Een goed data-transmissie systeem moet aan een aantal eisen voldoen, die te maken hebben met:
- de snelheid;
- de voedingen;
- de logische compatibiliteit;
- de drivers;
- de receivers;
- de lijnen.

In de volgende paragraafjes worden deze aspecten nader toegelicht.

Snelheid
De snelheid van de data-overdracht moet hoog genoeg zijn voor de geleverde dienst. Op dit moment zijn 20 kbps (20 duizend bit/seconde) voor single-ended systemen en 10 Mbps voor gebalanceerde systemen normale eisen.

Voedingen
Het gebruik van een enkele voedingsspanning van +5 V heeft de voorkeur, maar soms zijn dubbele spanningen, zoals +/-5 V, +/-9 V of +/-12 V nodig.

Logische compatibiliteit
Hieronder wordt over het algemeen "compatibel met TTL" verstaan, hoewel veel schakelingen ook compatibel zijn met low-power/low-level CMOS.

Drivers
Drivers zijn schakelingen die digitale signalen versturen, die dus geschikt zijn voor het aansturen van laag-impedante transmissielijnen.
Zij moeten ook bestand zijn tegen hogere spanningen dan de V_{CC}-niveaus.

Receivers
De receivers ontvangen de door de drivers via de transmissielijnen verzonden digitale signalen. Deze moeten een goede ingangsgevoeligheid hebben, gewoonlijk minder dan 500 mV. Ook moeten zij ongevoelig zijn voor storingen bijvoorbeeld door gebruik te maken van common-mode rejectie of het inbouwen van een hysteresis.

6.21 Het transport van digitale signalen

Lijnen
De data-transmissielijnen moeten over de gehele lengte uniforme impedantiekarakteristieken hebben, zodat afsluiting met de juiste impedantie hoge snelheden zonder reflecties garandeert.

Twee soorten data-overdracht
Data-overdracht kan op twee manieren plaatsvinden:
- single-ended;
- differentieel.

Bij single-ended data-transport wordt het signaal ten opzichte van aarde (de nul van de voeding) getransporteerd. Bij differentieel data-transport vindt de signaaloverdracht gebalanceerd plaats: er worden dan een signaaldraad plus een signaal-retourdraad gebruikt die beide "los van aarde" zijn.

Single-ended transmissie
De voordelen hiervan zijn:
- eenvoudig;
- goedkoop.

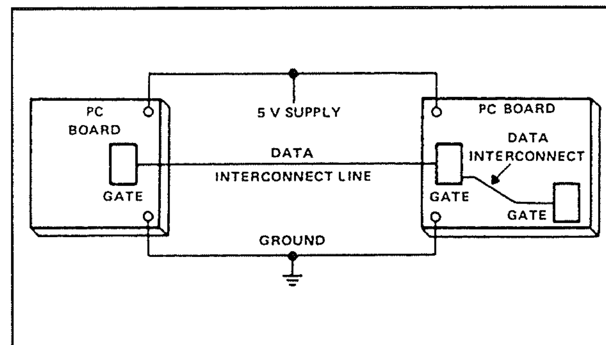
Het systeem heeft echter ook nadelen, zoals:
- zendt gemakkelijk stoorsignalen uit;
- storingsgevoelig (wordt met coax verbeterd, maar is dan wel duur);
- beperkte afstanden en data-snelheden.

Differentiële (gebalanceerde) transmissie
De voordelen van differentiële transmissie zijn:
- hoge common-mode storingsonderdrukking;
- minder uitstraling en RFI;
- hogere data-snelheden mogelijk;
- grotere afstanden mogelijk.

Het systeem heeft toch ook wat nadelen, zoals:
- duurder dan single-ended transmissie;
- getwiste paren of andere typen gebalanceerde transmissielijnen zijn nodig.

Typen transmissielijnen

Inleiding
Het zal duidelijk zijn dat de verbindingslijnen voor het datatransport een zeer belangrijke rol spelen. Er kan (afgezien van optische glasvezel-verbindingen) een keuze gemaakt worden uit:
- enkele draden;
- getwiste aderparen;
- afgeschermde getwiste paren;
- coaxiale kabel;
- platte bandkabel.

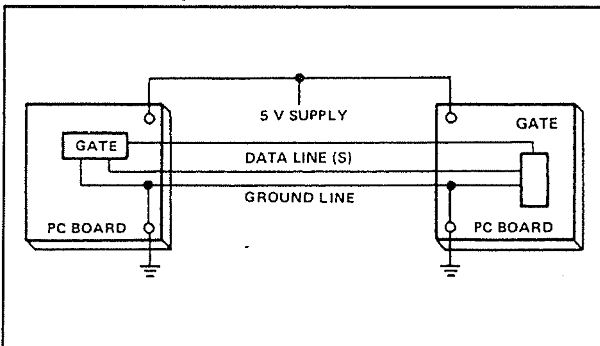


Figuur 3/6.21-2: Een enkele signaaldraad tussen zender en ontvanger. De retourverbinding van het signaal is GROUND.

Enkele lijn plus aarde
Het eenvoudigst en meest kwetsbaar is de enkele draad verbinding tussen driver en receiver, waarbij de aarde (nul) als retourleiding voor het signaal werkt, zie figuur 3/6.21-2. In sommige gevallen, voornamelijk bij korte afstand gecombineerd met lage transmissie-snelheden, kan deze oplossing ruim voldoende zijn. Het is hierbij zeer belangrijk dat de weerstand van de

6.21 Het transport van digitale signalen

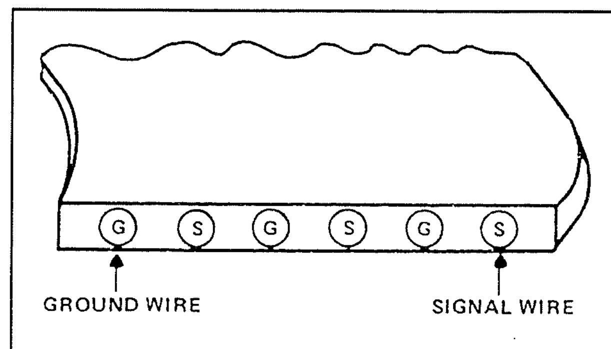
aardverbinding klein is. Is de ontvanger bijvoorbeeld een standaard TTL-poort, dan zal de ingangsstroom 1,6 mA bedragen. Omdat deze stroom ook door de retourleiding loopt, zal hierover een spanningsverlies optreden. Wanneer deze retourleiding een weerstand van 500 Ω heeft, zal dit spanningsverlies al 800 mV bedragen. Dit betekent dat het signaal aan de ingang van de ontvanger altijd als HOOG wordt gezien!



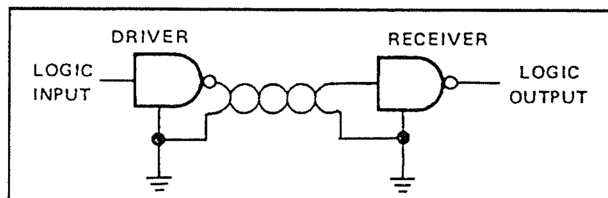
Figuur 3/6.21-3: Een multi-signaalleiding tussen zender en ontvanger. De retourverbinding van de signalen is nog steeds de massa.

Multi-lijn plus aarde

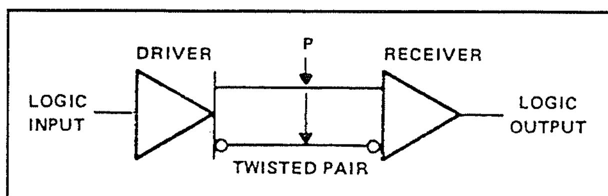
Binnen apparaten worden vaak niet-getwiste multi-lijn verbindingen volgens figuur 3/6.21-3 gebruikt om parallelle data bijvoorbeeld van de ene print naar de andere te transporteren. Elke signaalleiding zorgt hierbij voor de overdracht van informatie, terwijl één gemeenschappelijke retourleiding wordt gebruikt. Zoals hierboven al werd vermeld, is het van groot belang dat de weerstand hiervan klein genoeg is. Worden bijvoorbeeld 8 bit data parallel getransporteerd en zijn die allemaal LAAG, dan is de som van de ingangsstromen (bij standaard TTL) 12,8 mA en bedraagt het spanningsverlies over een weerstand van 100 Ω dus 1,28 V. Dit is ontoelaatbaar hoog. Het spanningsverlies kan op twee manieren worden verlaagd: door de weerstand van de retourleiding te verlagen en/of door de ingangsstroom van de ontvanger te verkleinen. De meeste lijn-receivers hebben daarom een zeer hoge ingangsimpedantie, waardoor de ingangsstroom per kanaal slechts enkele tientallen of honderdtallen µA bedraagt.



Figuur 3/6.21-4: Een platte bandkabelverbinding met gesplitste retourverbinding van de signalen: tussen twee signaaladers zit een massa (G).

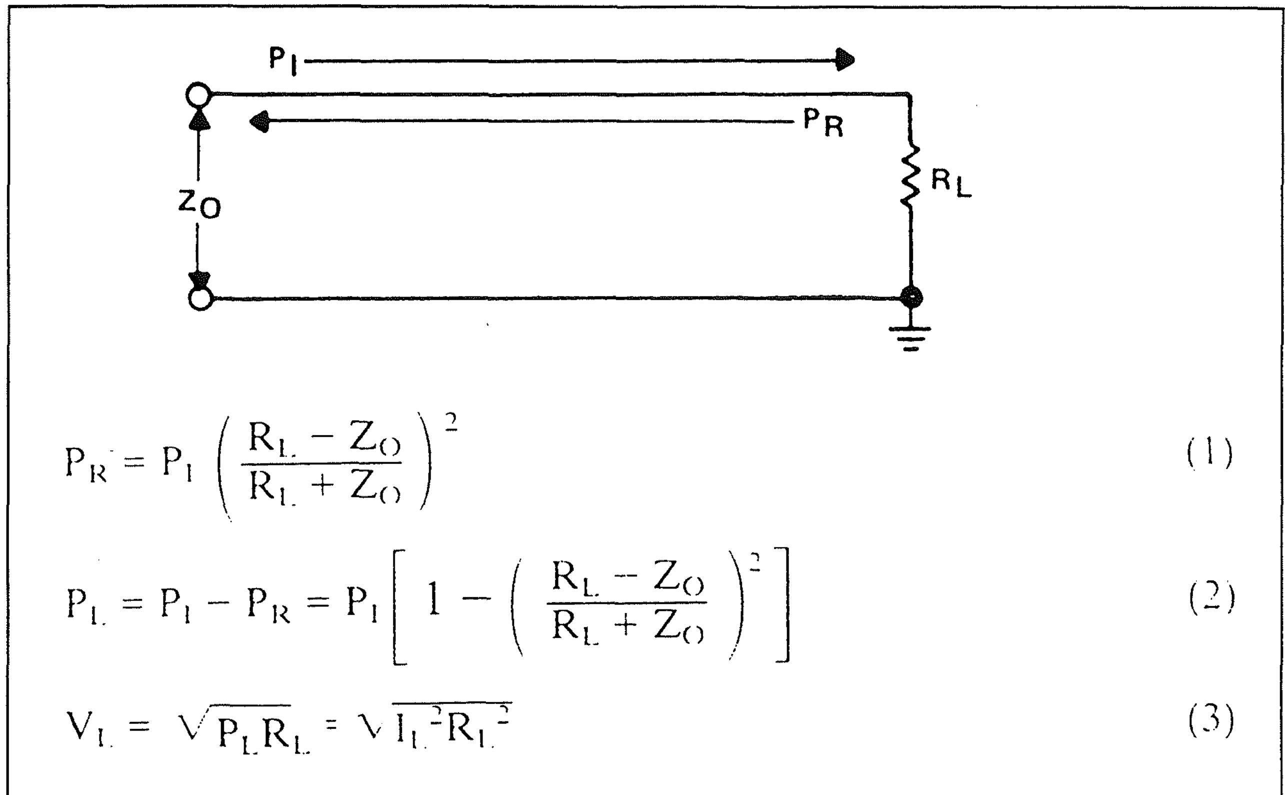


Figuur 3/6.21-5: Signaal-overdracht tussen zender en ontvanger met behulp van getwiste paren.



Figuur 3/6.21-6: Gebalanceerde (differentiële) data-overdracht met behulp van getwiste paren.

6.21 Het transport van digitale signalen



Figuur 3/6.21-7: De schematische voorstelling en bijbehorende formules van een transmissielijn met toegevoerd vermogen (Pi), gereflecteerd vermogen (Pr), afsluit-impedantie (RL) en karakterisitieke lijn-impedantie (Zo).

Platte bandkabel
Een variatie op deze multi-lijn verbinding is de platte bandkabel, voorgesteld in figuur 3/6.21-4. Deze wordt, vanwege het gemak, veel in computers gebruikt.
Deze verbinding kan sterk worden verbeterd door data en retourleidingen om-en-om aan te sluiten. Wanneer de datalijnen dan ook nog correct worden afgesloten, zijn relatief hoge data-snelheden mogelijk zonder dat reflecties of uitslingeringen optreden.

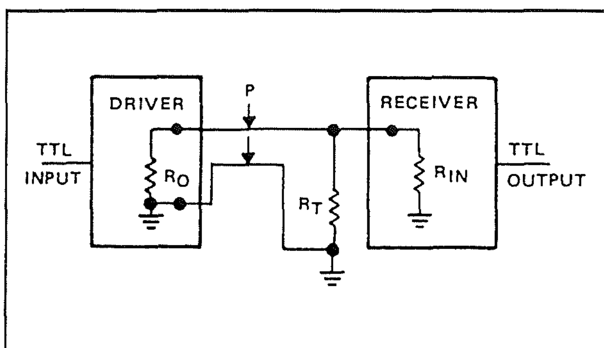
Getwiste paren
Hetzelfde geldt voor de verbinding met getwiste paren (figuur 3/6.21-5) die overigens ook zeer geschikt zijn voor gebalanceerde transmissie. Hierbij zijn beide draden "los van aarde" en werken ze beide als signaallijn (figuur 3/6.21-6).

Eigenschappen van verbindingen

Lijn-afsluiting en reflectie
De spanning over de impedantie waarmee een transmissielijn wordt afgesloten, is afhankelijk van de reële en de imaginaire bestanddelen van die impedantie, de karakteristieke impedantie van de lijn en het toegevoerde vermogen. Als de afsluitende impedantie zuiver Ohms is (in de praktijk kunnen de reactieve bestanddelen hiervan inderdaad worden verwaar-

6.21 Het transport van digitale signalen

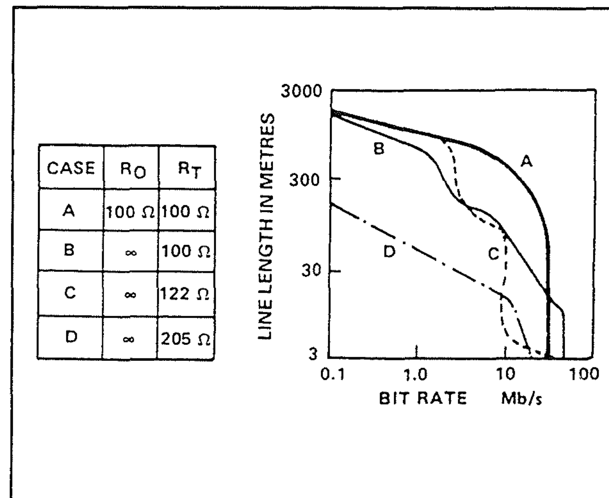
loosd) zijn de formules in figuur 3/6.21-7 geldig voor de transmissielijn.
Wanneer RL = Zo worden de tellers van de formules 1 en 2 nul en is het gereflecteerde vermogen dus nul. Hierdoor wordt een belangrijke bron van signaalvervorming en storingen geëlimineerd en wordt alle vermogen aan de belasting overgedragen (Pl = Pi).



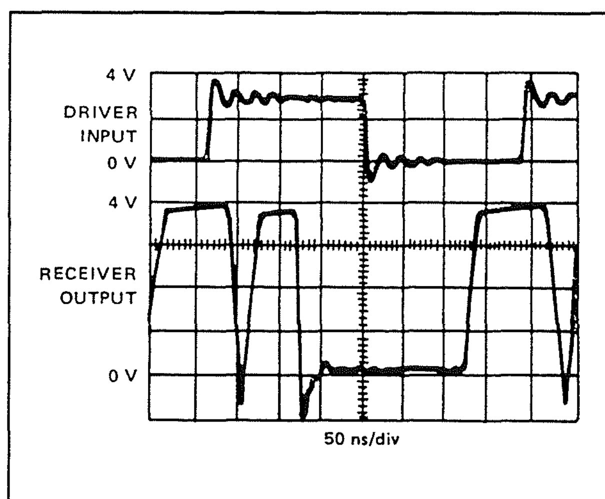
Figuur 3/6.21-8: Een zogenoemde "Single-ended" lijn-afsluiting.

Het afsluiten van de lijn

In de praktijk bestaat Rl uit de parallelschakeling van afsluitweerstand (Rt) en ingangsweerstand van de lijn-ontvanger (Rin) (zie figuur 3/6.21-8). Wanneer Rin>>Rt, worden grote geïnduceerde storingen door Rt naar aarde afgevoerd, waardoor de ingang van de ontvanger wordt beveiligd. Zoals in figuur 3/6.21-9 te zien is, hebben de waarden van Ro en Rt een grote invloed op het gedrag van de schakeling. Hierin wordt een 100 Ω getwist paar als datalijn gebruikt. In geval A wordt de lijn correct afgesloten en worden de beste prestaties geleverd. De onregelmatige vormen in de andere gevallen zijn het gevolg van reflecties.
In figuur 3/6.21-10 is het resultaat te zien van signaaloverdracht in geval C, wanneer de lijnlengte 15 m bedraagt en de datasnelheid 5,4 Mbps. Hoewel de bit-snelheid bij deze verkeerd afgesloten verbinding hoog kan zijn, ontstaan door de reflecties uitslingeringen op het signaal die verkeerd triggeren van de ontvanger tot gevolg kunnen hebben.



Figuur 3/6.21-9: De effecten van verschillende lijn-afsluitingen op de toegestane snelheid over de verbinding.

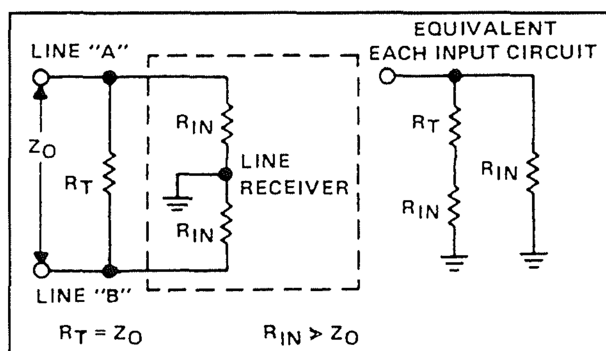


Figuur 3/6.21-10: Oscillogrammen van in- en uitgangssignaal van de ontvanger in geval C.

Stoorsignalen en interferenties

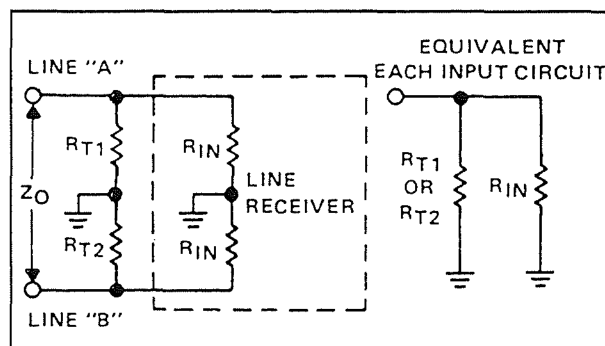
Stoorsignalen en interferenties die verlies van data of verkeerd gelezen data tot ge-

volg hebben, kunnen op vele manieren ontstaan. Stoorsignalen, afkomstig van elektromagnetische of elektrostatische bronnen, wekken interferentiespanningen op die evenredig zijn met de impedantie tussen lijn en aarde. Bij korte verbindingen met lage interferentieniveaus worden gebalanceerde lijnen meestal direkt met Rt van lijn A naar lijn B afgesloten (figuur 3/6.21-11). De waarde van de afsluitweerstand (Rt) moet gelijk zijn aan de karakteristieke impedantie van de lijn (Zo), wanneer mag worden aangenomen dat de ingangsweerstand van de ontvanger (Rin) veel groter is dan Zo. Zoals uit het vervangingsschema in figuur 3/6.21-11 blijkt, is (door Rin) de weerstand van elke lijn naar aarde zeer groot. Hierdoor zullen hoge stoorspanningen worden geïnduceerd.



Figuur 3/6.21-11: Een eenvoudige lijn-naar-lijn afsluiting.

Bij de afsluit-techniek van figuur 3/6.21-12 wordt de afsluitweerstand van de lijnen in tweeën gesplitst in waarden van 1/2*Zo. In dit geval is de weerstand naar aarde klein (ongeveer Rt1 of Rt2), waardoor de resulterende common-mode spanning veel lager is dan bij het schema van figuur 3/6.21-11. Een getwiste paarverbinding heeft meestal een karakteristieke impedantie van ongeveer 120 Ω, zodat Rt1 = Rt2 = 62 Ω.

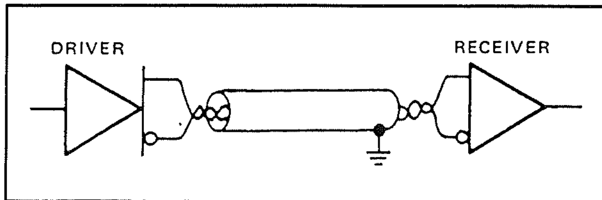


Figuur 3/6.21-12: Verbeterde lijn-afsluiting door de afsluitwerstand in twee even grote delen te splitsen.

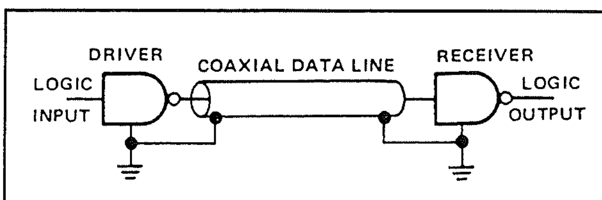
Common-mode spanning

Common-mode spanningen zijn spanningen die op de twee aders van een verbinding in even grote mate aanwezig zijn. Hoewel door gebalanceerde getwiste paarlijnen opgepikte storingen aanzienlijke stoorspanningen kunnen ontstaan, leveren die weinig problemen op omdat ze voor beide lijnen hetzelfde zijn. Door differentiële drivers en receivers worden common-mode spanningen onderdrukt. Deze berekenen immers het spannings*verschil* tussen de spanningen op beide aders en common-mode signalen leveren een verschilspanning op die gelijk is aan nul. Getwiste paren zijn verkrijgbaar als bandkabel, als aparte paren of als afgeschermde paren. Door afscherming, mits goed geaard, worden storingen beter onderdrukt maar hierdoor wordt de gedistribueerde capaciteit ook veel groter, waardoor het signaal bij hoge frequenties sterk wordt verzwakt. Wanneer een afgeschermde, getwiste kabel wordt gebruikt, zal het signaal het minst worden verzwakt en zullen ongewenste signalen het best worden onderdrukt als de mantel alleen bij de ontvanger wordt geaard volgens figuur 3/6.21-13.

6.21 Het transport van digitale signalen



Figuur 3/6.21-13: Data-overdracht met behulp van een afgeschermd, getwist paar.



Figuur 3/6.21-14: Signaal-overdracht met een standaard coaxiale kabel.

Coaxiale kabels

De populaire 50 tot 200 Ω coaxiale kabels hebben een zeer uniforme en exacte impedantie en bieden de beste karakteristieken als transmissielijn voor single-ended toepassingen (figuur 3/6.21-14).
De voordelen van het gebruik van coaxiale kabels zijn:
- geringe verliezen;
- goede afscherming tegen elektromagnetische storingen.

De signaalstroom moet zowel door de binnenader als door de afscherming vloeien om goed afgeschermd te zijn. Aan beide einden van de kabel dient een goede afsluiting aanwezig te zijn, terwijl problemen met betrekking tot aardlussen alleen kunnen worden opgelost door binnen het dynamisch bereik van het single-ended systeem te blijven of door te isoleren (bijvoorbeeld met transformatoren of opto-isolatoren).

3/6.22

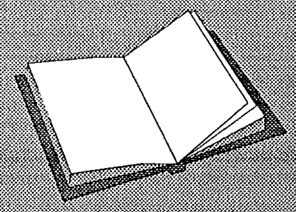
Werking en principes van Zero-Power SNAPHAT geheugens

Inleiding

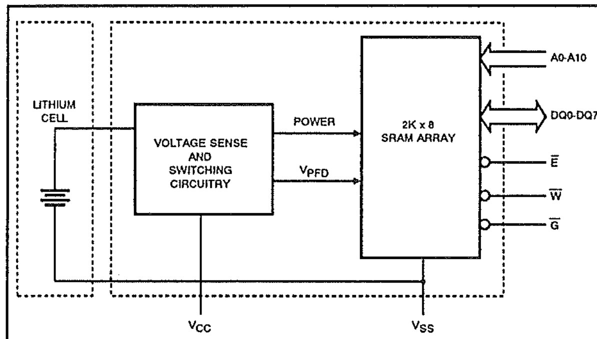
Het voedingsprobleem bij geheugens
Steeds meer apparatuur wordt bestuurd door microprocessoren of microcontrollers. Dergelijke apparatuur heeft dus steeds een stuk microcode, firmware genoemd, nodig die er voor zorgt dat de microprocessor of -controller aan het werk kan. Dergelijke gegevens kunnen in EPROM's ingebrand worden of eenmalig ingelezen in flash-geheugens. Echter, vaak moeten gedurende de werking gebruikersgegevens vastgelegd worden. Statische RAM's zijn uiteraard, vanwege hun eenvoudige adressering en data-behandeling, in principe de beste geheugens om dergelijke gegevens in op te bergen. Het probleem van een statisch RAM is echter dat de gegevens verloren gaan als de voedingsspanning wegvalt. Natuurlijk bestaat de mogelijkheid in het apparaat een batterij in de bouwen, die het geheugen onder spanning houdt als de voeding wegvalt. Dit betekent voor de ontwerper echter extra werk.

Zero-Power RAM's
Diverse halfgeleiderfabrikanten hebben dit probleem ingezien en bieden tegenwoordig zogenoemde "Zero-Power" RAM's aan. Dat zijn gewone RAM's, die echter voorzien zijn van een eigen batterijtje. Dit batterijtje neemt de spanningsvoorziening van het geheugen over als de voedingsspanning wegvalt. Naast de batterij en het eigenlijke geheugen zijn dergelijke IC's uiteraard voorzien van een extra schakeling, die de voedingsspanning van het IC controleert en, bij een te lage waarde, automatisch overschakelt naar de ingebouwde batterij. Het blokschema van een dergelijke Zero-Power RAM kan dus voorgesteld worden als getekend in figuur 3/6.22-1.
In de meeste gevallen wordt de spanning betrokken uit een kleine lithium-cel, die voldoende capaciteit heeft om data- en clock-functies gedurende minstens tien jaar te kunnen uitvoeren bij afwezigheid van externe voedingsspanning.

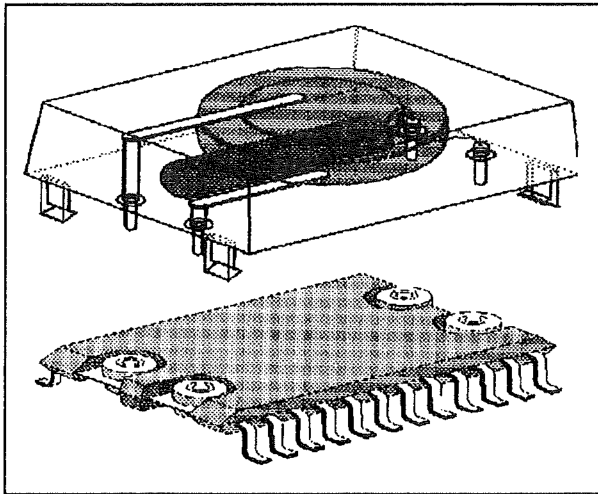
LEES OOK:
Hoofdstuk 3/3.23
Hoofdstuk 3/6.13
Hoofdstuk 3/6.15



6.22 Werking en principes van Zero-Power SNAPHAT geheugens



Figuur 3/6.22-1: Het blokschema van een Zero-Power RAM.



Figuur 3/6.22-2: De "SNAPHAT"-constructie van SGS Thomson.

SNAPHAT

Er zijn verschillende systemen verzonnen om de cel in de behuizing van het geheugen te integreren. Dergelijke cellen zijn echter nogal gevoelig voor oververhitting, hetgeen problemen kan opleveren als men het IC in de print soldeert. SGS Thomson heeft daar een unieke oplossing voor verzonnen, die onder de naam "SNAPHAT" gepatenteerd is. Zoals uit figuur 3/6.22-2 blijkt, zit de lithium-cel in een "hoedje", dat na het solderen van het geheugen-IC op de behuizing van de chip geklikt kan worden. Hierbij zorgen vier pennetjes ervoor dat de noodzakelijke elektrische verbindingen tussen de chip en de cel automatisch tot stand komen. Op deze manier is men er zeker van dat de gevoelige lithium-cel niet beschadigd kan worden door de hoge temperaturen, die bij het soldeerproces kunnen optreden.

Voorbeelden

Overzicht

SGS Thomson levert een aantal Zero-Power statische RAM's met deze SNAPHAT-voorziening:

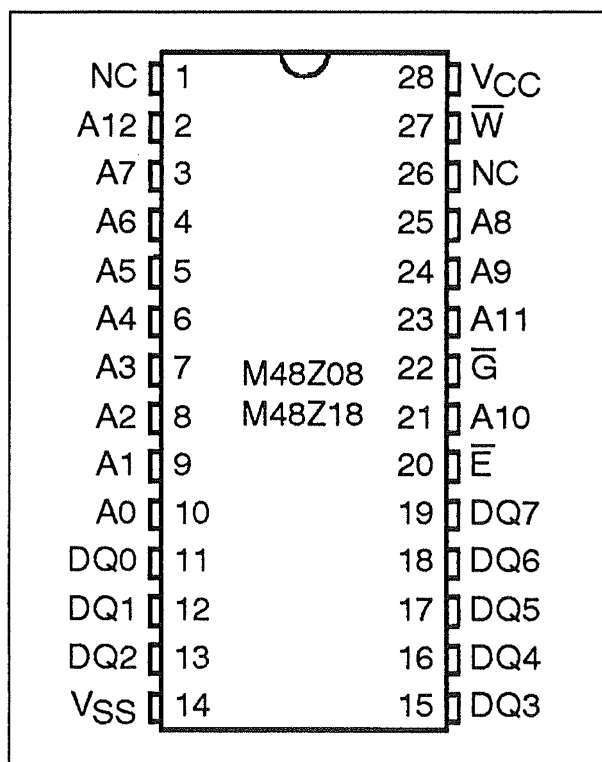
- M48Z08:
  8 k x 8, zowel voor DIL als voor SMD;
- M48Z18:
  8 k x 8, zowel voor DIL als voor SMD;
- M48Z35:
  32 k x 8, zowel voor DIL als voor SMD;
- M48Z58:
  8 k x 8, zowel voor DIL als voor SMD;
- M48Z59:
  8 k x 8, zowel voor DIL als voor SMD, met extra RESET-uitgang bij Power Fail.

Als voorbeeld worden één type in het kort besproken, de overigen werken identiek.

M48Z08

De M48Z08 is een 8 k x 8 bit niet-vluchtige statische RAM die pen- en functie-compatibel is met de MK48Z08. De monolithische chip is leverbaar in twee speciale behuizingen voor integratie met een lithium batterij om data gedurende tenminste elf jaar te kunnen vasthouden. De M48Z08 is een niet-vluchtige vervanger van elke willekeurige JEDEC-standaard 8 k x 8 SRAM en kan ook in vele ROM-, EPROM- en EEPROM-sockets geplaatst worden.

6.22 Werking en principes van Zero-Power SNAPHAT geheugens



Figuur 3/6.22-3: Aansluitgegevens van de M48Z08.

Net als bij een PROM wordt data vastgehouden, zonder beperking van het aantal schrijfcycli of speciale timing-eisen. De 28-pens 330-mil SO-behuizing heeft vergulde contacten om er een aparte SNAPHAT-behuizing die de batterij bevat op te plaatsen. De SNAPHAT kan maar op één manier worden gemonteerd. SO-behuizing en SNAPHAT worden apart geleverd.

De M48Z08 is voorzien van een Power-Fail detectie-schakeling die constant de 5 V voeding in de gaten houdt. Zodra V_{cc} buiten de specificaties gaat, wordt de schrijf-beveiliging ingeschakeld waardoor het geheugen geen data meer kan opnemen. Als V_{cc} lager wordt dan 3 V, wordt de batterij ingeschakeld om de data te behouden tot de voedingsspanning terugkomt.

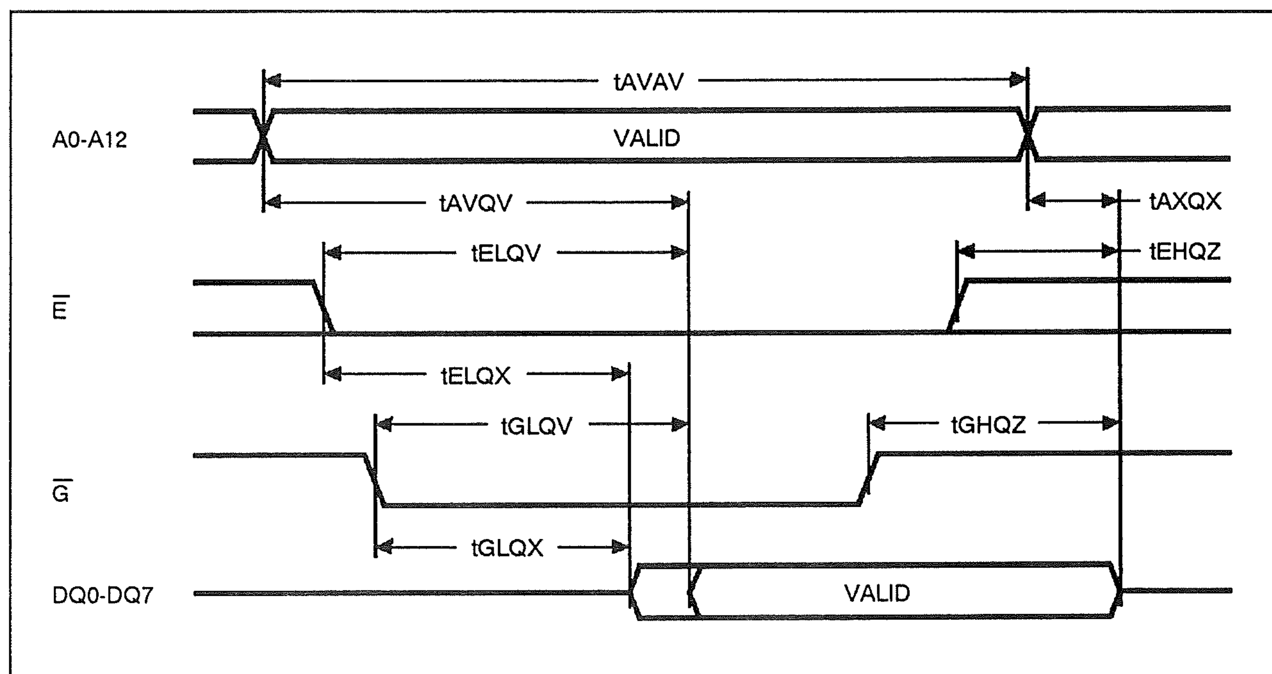
De aansluitgegevens van dit IC zijn getekend in figuur 3/6.22-3, het intern blokschema is gelijk aan dat in figuur 3/6.22-1, met dien verstande dat er 12 adreslijnen A0 tot en met A12 zijn en het array bestaat uit een matrix van 8 k x 8. De beveiliging tegen te lage voedingsspanning werkt met twee drempels, die V_{PFD} en V_{SO} worden genoemd. V_{PFD} is de “Write Protect”-spanning. Bij dit type is deze drempel gelegen tussen 4,5 V (inschakelen) en 4,75 V (weer uitschakelen). Daalt de beschikbare voedingsspanning onder de laagste waarde, dan komt de “Power Fail Deselect” automatisch in actie. De schrijfbeveiliging wordt ingeschakeld en de uitgangen worden hoogimpedant. De tweede drempel V_{SO} wordt “Switch Over”-spanning genoemd. Bij dit IC ligt deze waarde op 3,0 V. Daalt de voedingspanning onder deze grens, dan wordt de voeding afgeschakeld en neemt de interne cel de voeding van de geheugen-matrix over. Het IC wordt voor de rest inactief, maar de in het geheugen opgeborgen gegevens blijven bewaard. Uit deze gegevens kan de functie-tabel van de M48Z08 afgeleid worden, deze is gegeven in figuur 3/6.22-4. De M48Z08 komt in de leesmode als $\overline{W}$ (Write Enable) HOOG is en $\overline{E}$ LAAG is. Met behulp van de 13 adreslijnen zijn dan de 8.192 bytes aan data direct bereikbaar. Geldige data is binnen t_{AVQV} (Address Access Time) na het stabiel worden van het laatste adressignaal op de data I/O-pennen aanwezig, mits aan de toegangstijden van $\overline{E}$ en $\overline{G}$ wordt voldaan. Als niet aan de timing van $\overline{E}$ en $\overline{G}$ wordt voldaan, is er pas geldige data na t_{ELQV} (Chip Enable Access Time) of t_{GLQV} (Output Enable Access Time), afhankelijk van welke van de twee het laatst komt. De toestand van de acht 3-state data I/O-signalen wordt bepaald door $\overline{E}$ en $\overline{G}$.

6.22 Werking en principes van Zero-Power SNAPHAT geheugens

Mode	V_{CC}	$\overline{E}$	$\overline{G}$	$\overline{W}$	DQ0-DQ7	Power
Deselect	4.75V to 5.5V or 4.5V to 5.5V	V_{IH}	X	X	High Z	Standby
Write		V_{IL}	X	V_{IL}	D_{IN}	Active
Read		V_{IL}	V_{IL}	V_{IH}	D_{OUT}	Active
Read		V_{IL}	V_{IH}	V_{IH}	High Z	Active
Deselect	V_{SO} to V_{PFD} (min)	X	X	X	High Z	CMOS Standby
Deselect	$\leq V_{SO}$	X	X	X	High Z	Battery Back-up Mode

Note: X = V_{IH} or V_{IL}

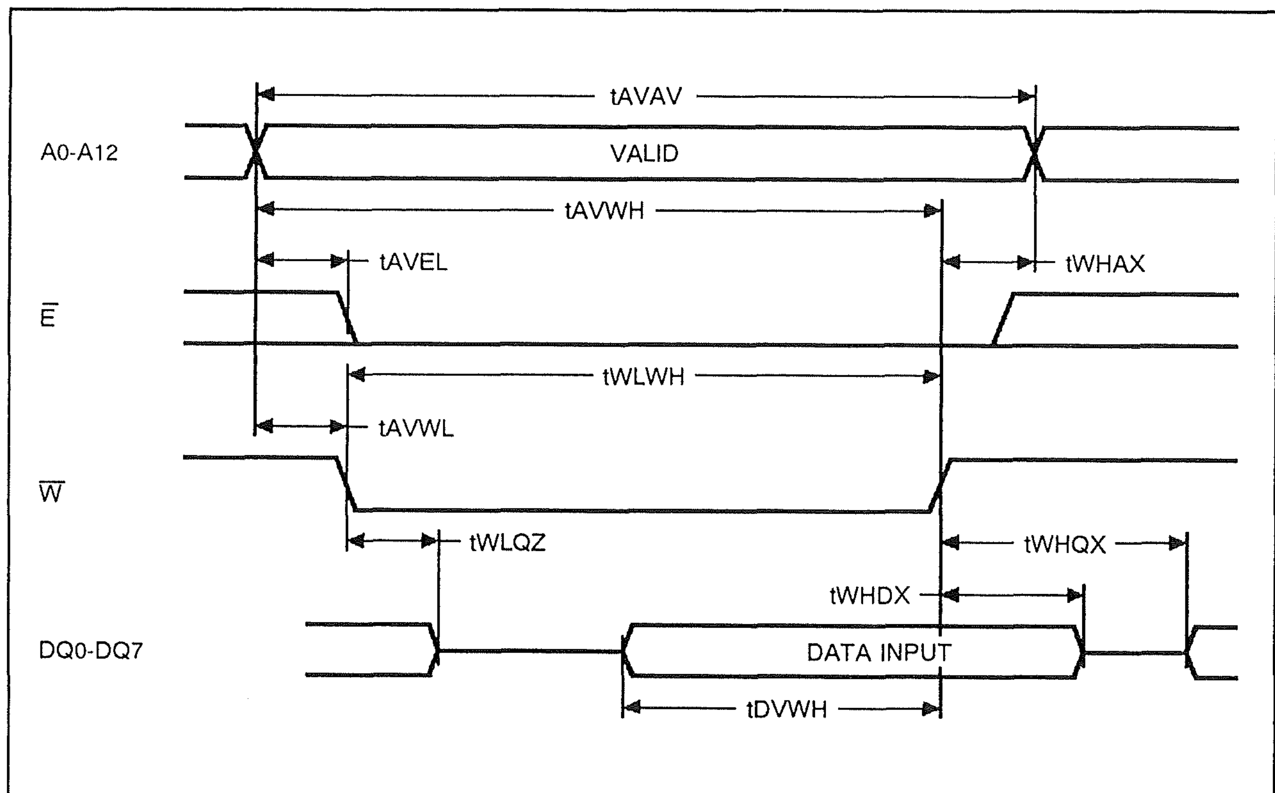
Figuur 3/6.22-4: De functie-tabel van de M48Z08.



Figuur 3/6.22-5: De timing bij het lezen van gegevens.

Als de uitgangen actief worden vóór t_{AVQV}, zijn de signalen onbepaald. Veranderen de adres-signalen terwijl $\overline{E}$ en $\overline{G}$ nog actief zijn, dan blijft de data nog geldig gedurende t_{AXQX} (Output Data Hold Time). De timing van de signalen die van belang zijn bij het lezen van de opgeslagen gegevens is weergegeven in figuur 3/6.22-5. Als $\overline{W}$ en $\overline{E}$ actief zijn bevindt de M48Z08 zich in de schrijfmode. Schrijven kan zowel geactiveerd worden met het $\overline{W}$-signaal (write enable) als met het $\overline{E}$-signaal (chip-enable). De schrijfcyclus begint op de laatst optredende achterflank van $\overline{W}$ of $\overline{E}$, terwijl het schrijven stopt op de eerst optredende stijgende flank.

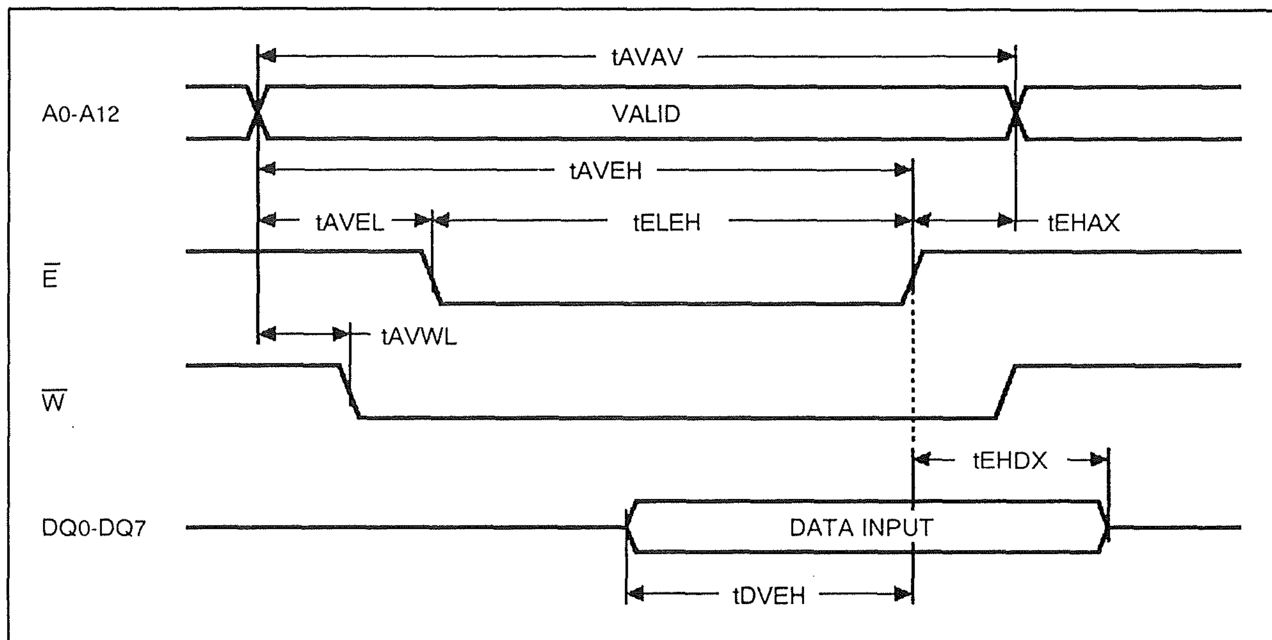
6.22 Werking en principes van Zero-Power SNAPHAT geheugens



Figuur 3/6.22-6: Timing-diagram bij een schrijf-cyclus die geïnitialiseerd wordt door Write Enable $\overline{W}$.

Het adres moet gedurende de gehele cyclus geldig zijn. Voorafgaande aan het begin van een nieuwe schrijfcyclus moet $\overline{E}$ of $\overline{W}$ minimaal gedurende t_{EHAX} na Chip Enable of t_{WHAX} na Schrijf Enable HOOG gaan. Data-in moet minimaal t_{DVWH} vóór het einde van de schrijfcyclus geldig zijn en tot t_{WHDX} geldig blijven. Tijdens de schrijfcycli dient $\overline{G}$ HOOG te blijven om busconflicten te vermijden.
In de figuren 3/6.22-6 en 3/6.22-7 zijn de timingen voor beide schrijf-acties samengevat. Wanneer een bruikbare V_{cc} is aangelegd, werkt de M48Z08 als een conventionele "byte-wide" statische RAM. Als de voedingsspanning daalt komt de "Power-Fail Deselect" automatisch in actie, waarbij de schrijf-beveiliging in het V_{cc}-gebied tussen $V_{PFD(max)}$ en $V_{PFD(min)}$ inschakelt. Alle uitgangen worden dan hoog-impedant en de ingangen "don't care". Komt V_{cc} beneden V_{SO}, dan wordt de interne batterij ingeschakeld om de data te behouden en de clock te bekrachtigen. Als de power-fail tijdens een schrijfcyclus optreedt, kan wel de data op het dan geldende adres beschadigd raken, maar niet de overige data in het geheugen. Het wordt aanbevolen V_{cc} te ontkoppelen. Wanneer V_{cc} weer boven V_{SO} uitkomt wordt V_{cc} op de RAM aangesloten, terwijl de batterij wordt losgekoppeld.
De schrijf-beveiliging blijft ingeschakeld totdat V_{cc} hoger is dan $V_{PFD(max)}$. Gedurende deze tijd moet $\overline{E}$ HOOG blijven om onbedoeld schrijven te voorkomen.

6.22 Werking en principes van Zero-Power SNAPHAT geheugens



Figuur 3/6.22-7: Timing-diagram van een door Chip Enable $\overline{E}$ gestuurde schrijf-cyclus.

3/6.23

Werking en principes van clock-drivers/generatoren

Theoretische achtergronden

Inleiding

Door de steeds maar toenemende snelheid van microprocessors en -controllers is de timing daarvan zeer kritisch geworden. De zogenaamde geheugenbandbreedte is in korte tijd al met een factor 10 gegroeid tot ruim 100 MHz en neemt nog steeds toe. Pentium II processors kunnen alleen nog van synchrone DRAM's gebruik maken. Bij de aansturing van grote SDRAM-banken is een goede werking niet alleen afhankelijk van de nauwkeurige gelijkloop, de vorm en de belastbaarheid van de clock-signalen, maar ook van de layout van de sporen op de print.

Bij de steeds hogere clockfrequenties (33, 50, 66, 100 MHz en nog hoger) wordt de onzekerheid of "skew" in de clock-distributie een steeds groter probleem. Aangezien de clock-signalen zowel worden gebruikt voor de aandrijving van de processors als voor het synchroniseren van het data-transport tussen de systeem-eenheden worden aan de clock-distributie hoge eisen gesteld. Als bij het ontwerpen van een clock-distributiesysteem de skew buiten beschouwing blijft, leidt dat onherroepelijk tot een onbetrouwbaar systeem.

Twee soorten clock-drivers

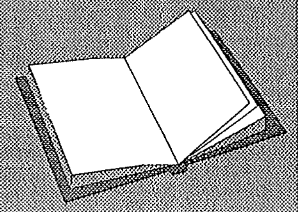
Er zijn twee soorten clock-drivers: buffers en op de PLL-technologie (Phase-Locked Loop) gebaseerde schakelingen.

De skew is bij buffers (goedkoop en eenvoudig) het grootst en wordt vooral veroorzaakt door belastingsverschillen aan de uitgangen. In op PLL gebaseerde schakelingen is de skew meestal zeer klein, aangezien de belastingsverschillen bij zo'n schakeling kunnen worden gecompenseerd. SDRAM-array's worden vaak door de PLL-schakelingen aangestuurd, waarbij dan naast skew ook "jitter" kan optreden.

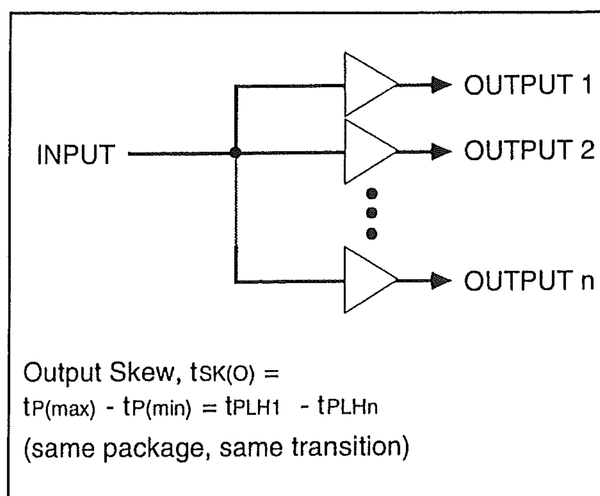
Wat is skew?

Met de term clock-skew worden de tijdverschillen in een clock-distributiesysteem beschreven. In figuur 3/6.23-1 is een voor-

LEES OOK:
Hoofdstuk 3/16



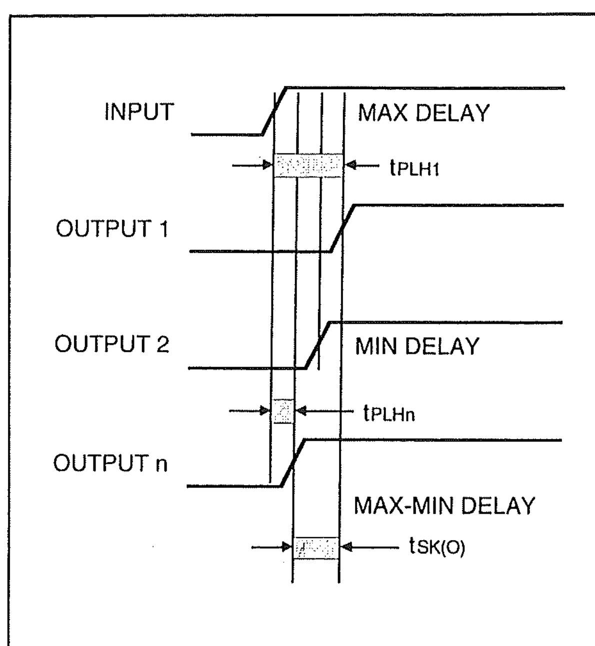
beeld te zien van een clock-driver met meerdere uitgangen en in figuur 3/6.23-2 de bijbehorende timing voor een LAAG-naar-HOOG overgang. Op alle ingangen wordt een gemeenschappelijk signaal gezet, waarvan n kopieën ontstaan. De clock-skew is het verschil in voortplantingsvertraging tussen de langzaamste en de snelste uitgang. Aangezien in figuur 3/6.23-2 uitgang n de kleinste vertraging heeft en uitgang 1 de grootste, is de clock-skew: $t_{PLH1} - t_{PLHn}$.



Figuur 3/6.23-1: Schema van een gebufferde clock-driver.

Er zijn meestal twee oorzaken van clock-skew in een systeem.

- **Skew van de driver**
  De eerste is de skew van de driver zelf. De clock-driver bevat interface-logika voor het aansturen van de clock-lijnen en is daardoor vanzelf al een oorzaak van skew. In een ideale clock-driver zijn alle interne componenten precies op elkaar afgestemd, zodat de voortplantingsvertragingen door gelijksoortige paden volkomen identiek zijn. In een praktische schakeling hebben veel variabelen invloed op de vertraging en dragen daardoor bij aan de skew: de layout en elektrische eigenschappen van de schakelelementen, de plaatsing daarvan ten opzichte van aarde en V_{CC} plus de paracitaire capaciteiten van de behuizing. Veel van deze variabelen zijn afhankelijk van het fabricageproces. Dit wordt **Output Skew** genoemd.

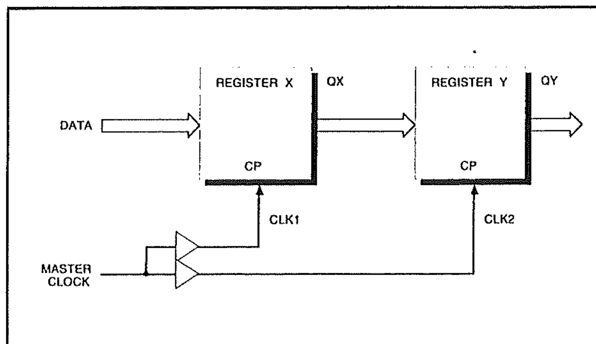


Figuur 3/6.23-2: Timing van de schakeling in figuur 3/6.23-1 voor het bepalen van de Output Skew $t_{SK(O)}$.

- **Clock-distributie systeem**
  De tweede oorzaak van clock-skew is het clock-distributie systeem. Het maakt veel uit hoe de clock-driver in het clock-distributie systeem is opgenomen. Hierbij zijn onder andere de layout van de signaallijnen, de belasting van de driver, de aansluitingen van de voedingslijnen en de ontkoppeling van de voeding van belang. Ook spelen bedrijfscondities, zoals de hoogte van de voedingsspanning en de omgevingstemperatuur een belangrijke rol. Van-

6.23 Werking en principes van clock-drivers/generatoren

wege de hoge schakelsnelheden bij de moderne high-speed logica is het verstandig om de sporen op een printkaart te beschouwen als transmissielijnen. De hierdoor optredende skew wordt ook wel **Board Skew** genoemd.



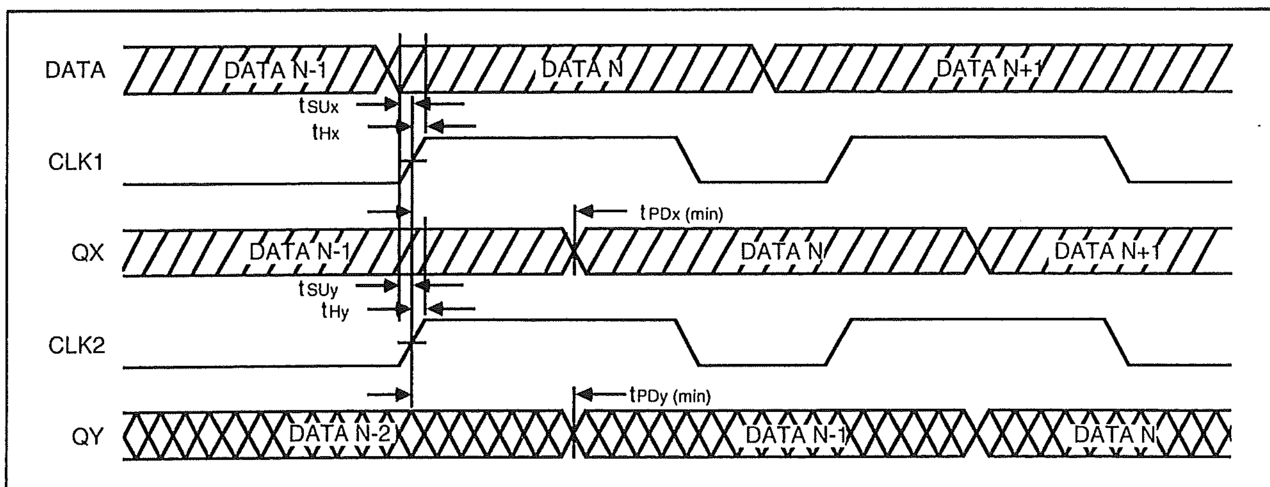
Figuur 3/6.23-3: Schema van een 2-register pijplijn.

Problemen door clock-skew

Als niet aan de vereiste timing van een systeem-element wordt voldaan zijn clock-skew problemen het gevolg. Veel van de gebruikelijke knelpunten vallen in twee categorieën. De eerste is het synchronisatie-probleem dat wordt veroorzaakt door skew tussen verschillende kopieën van een systeem-clock. De tweede is het probleem van het niet voldoen aan de door de systeem-componenten opgelegde duty-cycle.

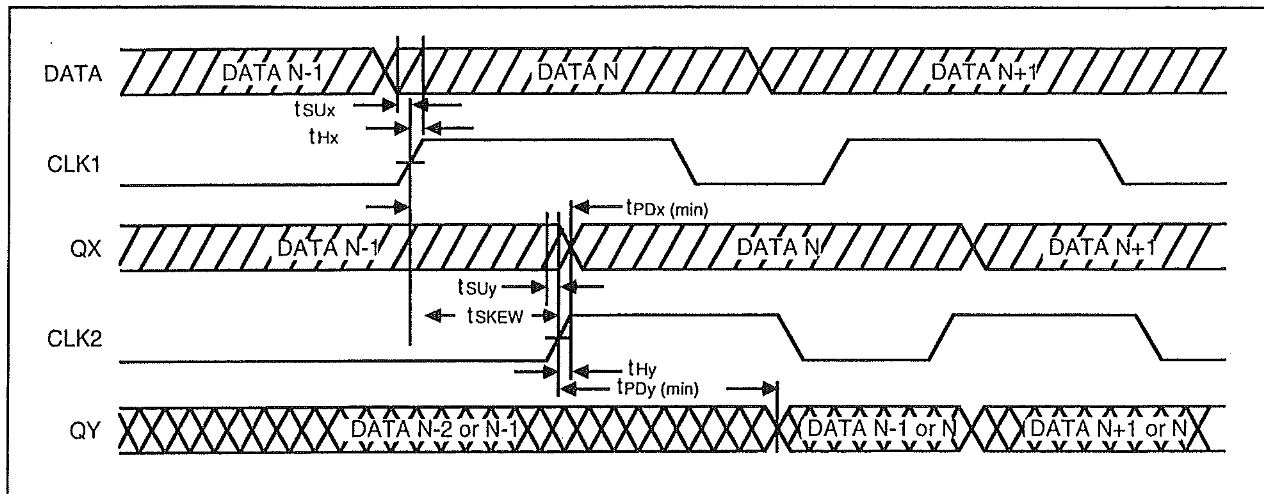
Het synchronisatie-probleem kan worden geïllustreerd aan de hand van een eenvoudig pijplijnregister, zie figuur 3/6.23-3. De pijplijn bestaat uit twee registers en wat clock-logika. De clock-schakeling begint met een Master-Clock die door buffers in tweeën wordt gedeeld (CLK1 en CLK2). CLK1 drijft register X aan en CLK2 register Y. De registers geven seriële data door bij elke clock-cyclus, waarbij de huidige uitgang van het Y-register overeenkomt met de vorige uitgang van het X-register. De timing hiervan is te zien in de figuren 3/6.23-4 en -5.

In figuur 3/6.23-4 is N het signaal aan de ingang van Register X en N-1 het ingangssignaal van Register Y. Voor een goede werking moet het ingangssignaal van elk register voldoen aan de vereiste setup- en houd-tijden ten opzichte van de clock. Aangezien de uitgang van Register X de ingang van Register Y is, mag de houd-tijd t_{Hy} niet langer zijn dan $t_{PDx(min)}$. In figuur 3/6.23-4 schakelen CLK1 en CLK2 op hetzelfde moment zodat de uitgang van Register X voldoet aan de voor Register Y vereiste setup- en houd-tijden.



Figuur 3/6.23-4: Pijplijn-timing zonder skew tussen CLK1 en CLK2.

6.23 Werking en principes van clock-drivers/generatoren



Figuur 3/6.23-5: Pijplijn-timing met skew tussen CLK1 en CLK2.

In figuur 3/6.23-5 is CLK2 vertraagd ten opzichte van CLK1 en ontstaat een t_{SKEW}. In dit geval mag t_{Hy} niet groter zijn dan $t_{PDx(min)}$ - t_{SKEW}. Met stelt vast dat de skew in CLK2 een fout in de vereiste houd-tijd van data N-1 veroorzaakt en ook in de setup-tijd van data N als ingang van Register Y. Om goed te werken moet data N-1 in Register Y geklokt worden, maar in figuur 3/6.23-5 is het onzeker of dat met data N of N-1 gebeurt. Als de tijdsmarge $t_{PDx(min)}$ - t_{Hy} ongeveer 2,5 ns bedraagt, is een clock skew van 2,5 ns of meer funest voor de betrouwbaarheid van het systeem. Bij veel microprocessor-systemen is het ook nodig dat de clock een bepaalde duty-cycle heeft. Bij standaard interface-logica kan een vaste duty-cycle bij hoge clocksnelheden echter moeilijk worden gegarandeerd omdat de voortplantingsvertragingen voor LAAG-naar-HOOG en HOOG-naar-LAAG overgangen zelden identiek zijn.

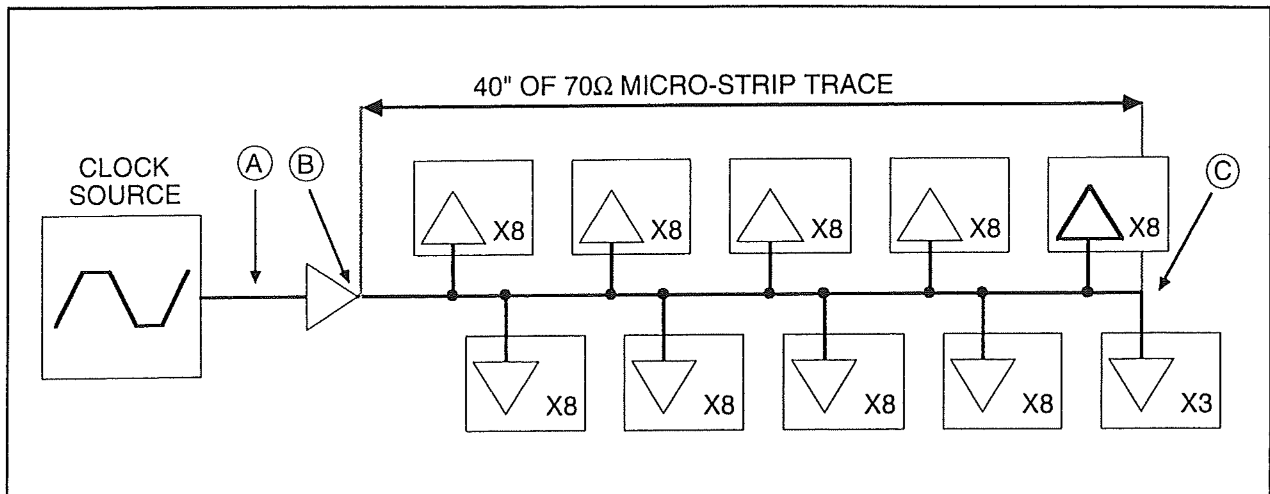
Ook zijn de tijdverschillen tussen de overgangen niet evenredig met de frequentie. Als een driver bijvoorbeeld een puls-skew van 3,0 ns heeft, is de tolerantie van een 25 ns cyclustijd (40 MHz) 112 %. Wordt de clock-frequentie verhoogd tot 50 MHz, dan wordt de tolerantie 115 %. Als vuistregel geldt dat niet meer dan 10 % van de clock-cyclus aan distributie mag worden besteed. Het is duidelijk dat niet aan deze richtlijn kan worden voldaan als standaard logica bij hogere frequenties wordt toegepast.

Oplossingen

Vereenvoudiging van de clock-distributie
Om aan te tonen hoe speciale clockdrivers het ontwerp kunnen vereenvoudigen wordt als voorbeeld een theoretisch clock-distributiesysteem gekozen waarbij een 50 MHz clock-signaal 75 belastingen moet aansturen. Elke belasting is een CMOS-ingang die verbonden is door een microstrip met een dichtheid van 1 belasting per 0,5 inch. Verder wordt verondersteld dat alle ingangen op de stijgende flank worden getriggerd en dat de skew minimaal moet zijn.

Eén manier zou zijn om alle 75 ingangen op een enkele clock-driver aan te sluiten (zie figuur 3/6.23-6). Dit levert echter veel problemen op.

6.23 Werking en principes van clock-drivers/generatoren



Figuur 3/6.23-6: Distributiesysteem met één enkele clock-driver en 75 belastingen.

Ten eerste is er de grote capaciteit van 75 CMOS ingangen. Als elke ingang een capaciteit van 10 pF heeft, is de totale capacitieve belasting 750 pF. Een standaard clock-driver zoals de 74FCT244A heeft een t_{PLH} van 2 ns/100 pF voor belastingen van meer dan 50 pF. Alleen al door deze capaciteit neemt de vertraging toe met 14 ns. Worden de 75 belastingen verbonden door een enkel spoor, dan bedraagt de lengte daarvan 38" (75 x 0,5 ingangen/inch) en is de vertraging van punt B naar punt C 5,7 ns (38" x 0,15 ns/"). Wanneer de praktische waarde van 0,37 ns/" voor de vertraging van een belast spoor wordt gebruikt, komt de skew tussen de uiteinden van het spoor overeen met 15 ns (38" x 0,37 ns/"). Bij een cyclustijd van 20 ns (50 MHz) wordt met 14 ns clock skew 70 % van de clock-cyclus aan distributie besteed.

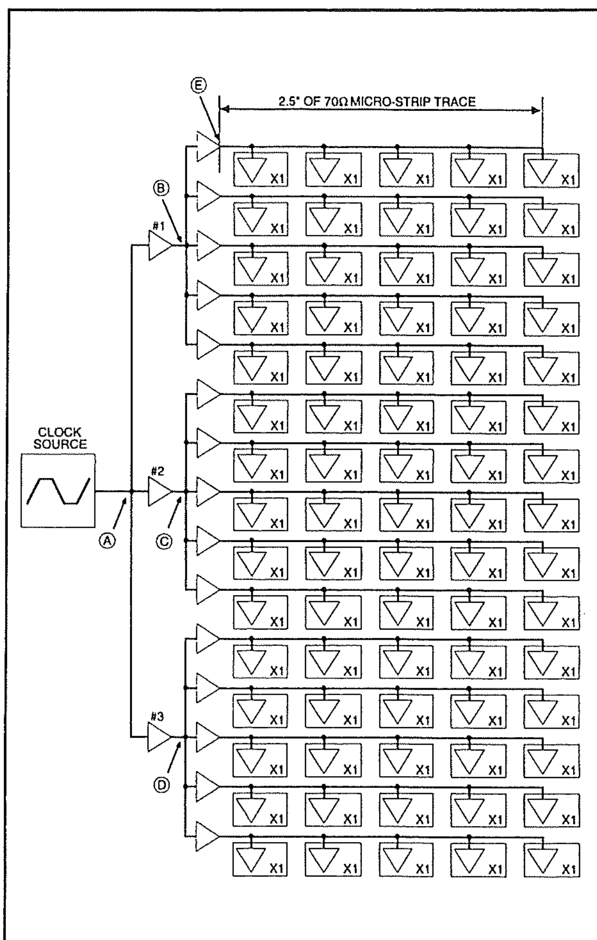
Clock-"tree"

Een tweede benadering is de clock-"tree" (zie figuur 3/6.23-7). Door een niveau van buffers in te voegen tussen de clock-bron en de 75 belastingen, neemt de capacitieve belasting van de buffer-uitgangen af van 750 pF tot 50 pF en wordt de spoorlengte van elke driver verminderd tot 2,5". Worden ook nu weer 74FCT244A's als driver gebruikt, dan zijn hier minstens 3 stuks van nodig (8 drivers per behuizing). Aangezien van de 244's geen skew wordt opgegeven, zou de ontwerper ervan kunnen uitgaan dat elke driver binnen een marge van 3,3 ns omschakelt (t_{PHLmax} - t_{PHLmin} = 4,8 - 1,5 ns). Als de uitgangen bij de punten B, C en D binnen 3,3 ns omschakelen, zal dat bij het tweede niveau (punt E) binnen een marge van 6,6 ns gebeuren.
Bij een cyclustijd van 20 ns (50 MHz) wordt dan 33 % van de cyclustijd aan distributie besteed (nog zonder de transmissielijn-effecten er in te betrekken).

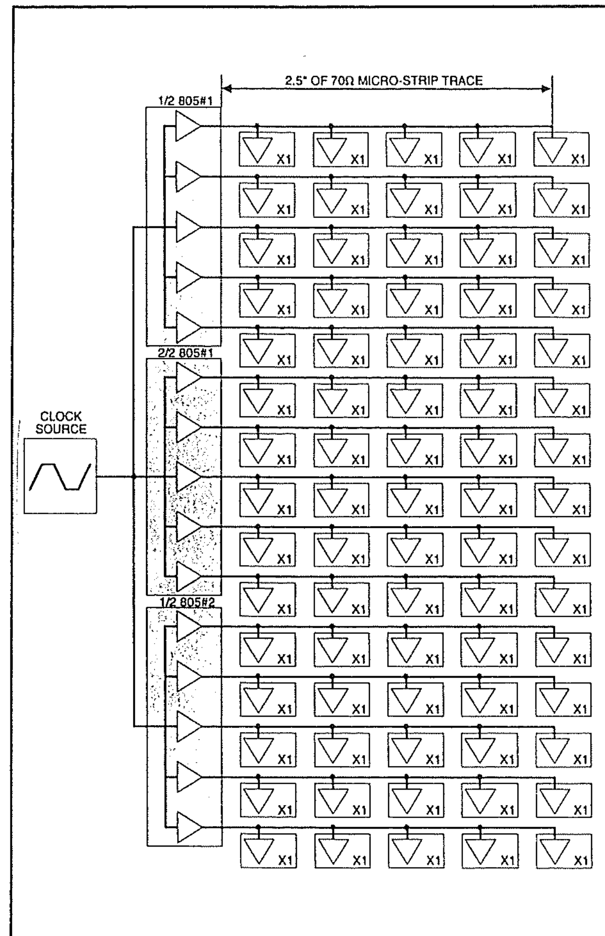
Speciale clock-drivers

De derde manier is het toepassen van speciale clock-drivers, zoals bijvoorbeeld de 49FCT805A van IDT (figuur 3/6.23-8). Aangezien elke groep van zes buffers in figuur 3/6.23-7 kan worden vervangen door een halve 49FCT805A, zijn hier slechts twee stuks van nodig. De skew van deze driver bedraagt slechts 1,5 ns zodat de skew-marge ten opzichte van figuur 3/6.23-7 afneemt van 6,6 ns tot 1,5 ns. Als

0,925 ns vertraging van een belast spoor in aanmerking wordt genomen (2,5" x 0,37 ns/") bedraagt de maximale skew nu 2,425 ns (inclusief een 1e orde behandeling van transmissielijn-effecten). Bij een cyclustijd van 20 ns wordt dan slechts 12 % van de tijd aan clock-distributie besteed. De 49FCT805A levert dus, vergeleken met het tweede ontwerp, een skew-verbetering van 77 % op. Bovendien is de belasting van de vorige trappen minder, zijn er minder IC's nodig en is de layout van de printkaart eenvoudiger.



Figuur 3/6.23-7: Distributiesysteem met 74FCT244's als clock-driver en 75 belastingen.

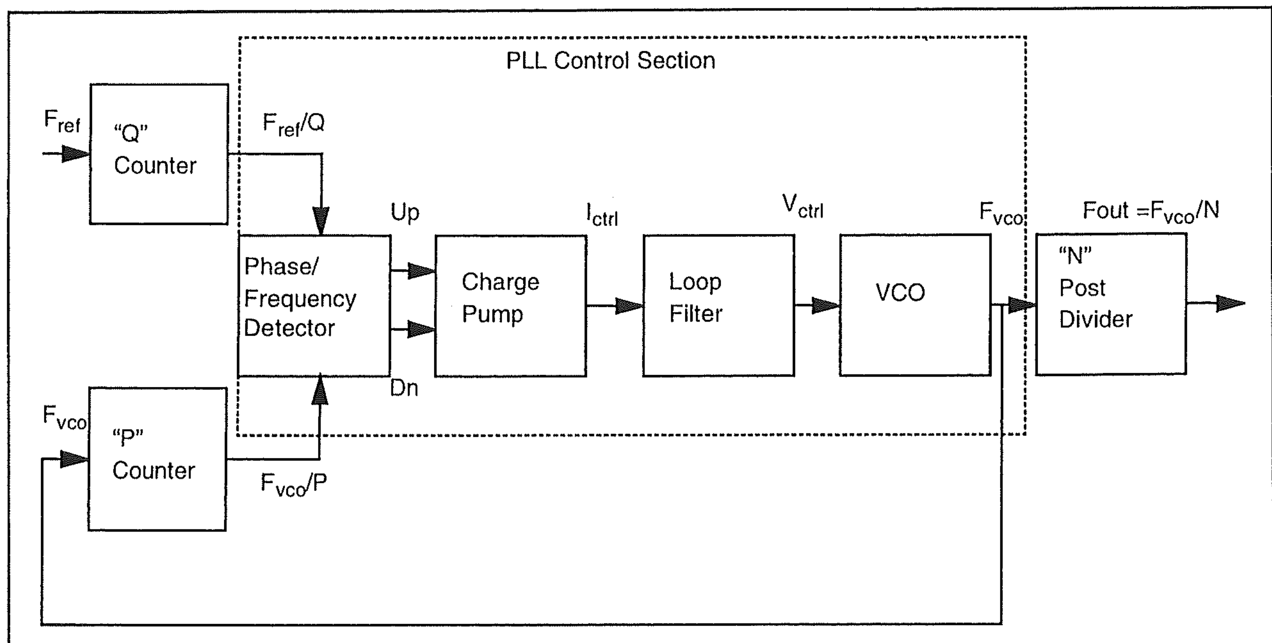


Figuur 3/6.23-8: Distributiesysteem met 49FCT805A's als clockdriver en 75 belastingen.

Op PLL gebaseerde frequentie-synthesizers

In frequentie-synthesizers worden één of meer Phase-Locked Loops (PLL) gebruikt om één of meer frequenties te genereren uit één of meer referentiebronnen. De referentie-frequentie is meestal afkomstig van een kristal dat op de synthesizer is aangesloten. Met een frequentie-synthesizer kunnen meerdere oscillatoren in een systeem worden vervangen, waardoor op ruimte en kosten kan worden bespaard. In figuur 3/6.23-9 is het blokschema van een PLL te zien.

6.23 Werking en principes van clock-drivers/generatoren



Figuur 3/6.23-9: Blokschema van een Phase-Locked Loop.

Een PLL heeft twee ingangen: een referentie-ingang en een feedback-ingang. Frequentie wordt op twee manieren gecorrigeerd. Grote afwijkingen in frequentie tussen de referentie-ingang en de feedback-ingang worden het eerst gecorrigeerd. Dit is de "ruwe" regeling van een PLL. De "fijne" regeling geschiedt door middel van fase-correctie.
De Phase/Frequency Detector in figuur 3/6.23-9 detecteert verschillen in fase en frequentie tussen de referentie- en de feedback-ingang en genereert Up- en Down-signalen. Als de frequentie op de feedback-ingang lager is dan de referentie-frequentie is de pulsbreedte van het Up-signaal groter dan van het Down-signaal (en vice versa). Deze besturingssignalen worden door een ladingspomp en een loop-filter geleid om een regelsignaal te vormen dat aan een spanningsgestuurde oscillator (VCO) wordt toegevoerd. De frequentie van deze oscillator is afhankelijk van de V_{ctrl}-ingang.

In rust is de VCO-frequentie:
$F_{VCO} = F_{ref} * P/Q$
zodat de uitgangsfrequentie van de PLL kan worden uitgedrukt als:
$F_{out} = (F_{ref} * P)/(Q * N)$
Hierin is:
- FVCO = VCO frequentie;
- Fref = referentie-frequentie;
- P = multiplier (is opgenomen in feedback-pad);
- Q = divider (is opgenomen in referentie-pad);
- N = post divider.

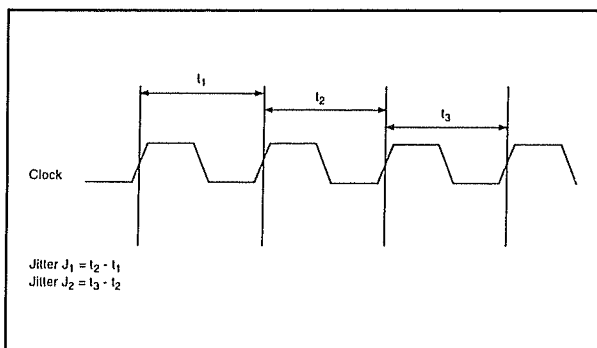
Jitter

Jitter
Jitter kan worden gedefinieerd als de afwijkingen in de tijd van de overgangen aan de uitgang van een clock ten opzichte van hun ideale posities. De afwijking kan zowel voor- als naijlen en wordt opgege-

ven in +/-picoseconden (ook wel in een percentage van de frequentie of absolute waarde: ns). Er zijn drie soorten jitter:
- cyclus-cyclus jitter
- periode jitter
- lange-termijn jitter (jitter wordt gemeten bij een gespecificeerde spanning).

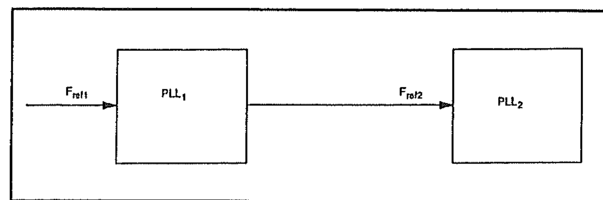
Cyclus-cyclus jitter

Onder cyclus-cyclus jitter verstaat men de verandering van een uitgangstransitie van de clock ten opzichte van de overeenkomstige positie in de vorige cyclus. Vanwege de korte tijden is deze soort jitter het moeilijkst te meten met een Timing Interval Analyzer. In figuur 3/6.23-10 is dit soort jitter te zien, waarin j_1 en j_2 de gemeten jitter-waarden zijn. De grootste waarde (gedurende meerdere cycli) is de maximum cyclus-cyclus jitter.



Figuur 3/6.23-10: Cyclus-cyclus jitter.

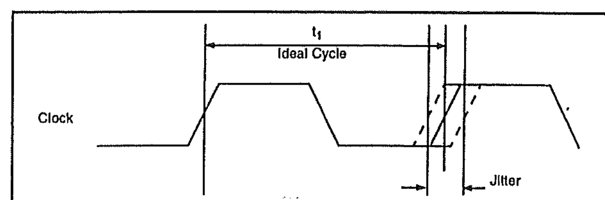
Vroeger (kort geleden) was cyclus-cyclus jitter meestal niet erg interessant. Maar nu de CPU's worden uitgerust met PLL's om de inwendige snelheid te verhogen (zoals bij de 486 en Pentium), is dat veranderd. In figuur 3/6.23-11 bijvoorbeeld is de uitgang van PLL1 de referentie van PLL2. Wanneer PLL2 niet in staat is te "locken" op de referentie-frequentie wordt dat waarschijnlijk veroorzaakt doordat de cyclus-cyclus jitter van PLL1 te groot is voor het "vang"-gebied van PLL2. Als PLL2 in een CPU is opgenomen moet de cyclus-cyclus jitter van PLL1 voldoende klein zijn.



Figuur 3/6.23-11: Voorbeeld van gekoppelde PLL's. PLL1 is hier de clock-generator, terwijl PLL2 zich bijvoorbeeld in een CPU bevindt.

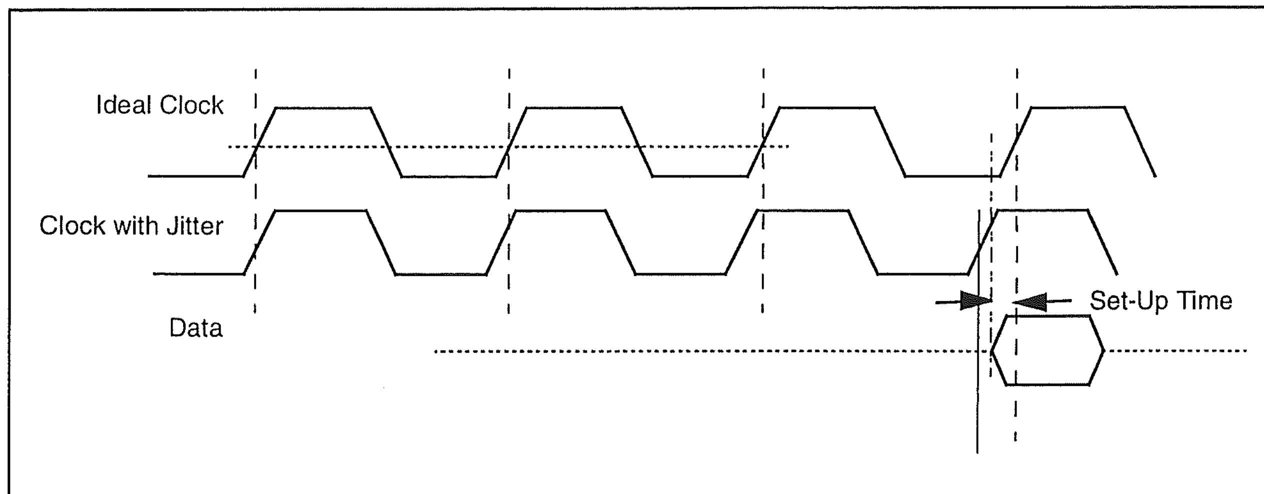
Periode jitter

Met periode jitter wordt de maximale afwijking van de uitgangstransitie ten opzichte van de ideale positie verstaan (zie figuur 3/6.23-12). Metingen van periode jitter worden gebruikt om de tijdslimieten in systemen te berekenen. Als bijvoorbeeld de processor in een systeem 2 ns data setup-tijd nodig heeft en de clock een maximale periode jitter van 2,5 ns heeft kan de stijgende flank van de clock al optreden voordat er geldige data op de databus staat. Omdat de processor hierdoor dan foute data krijgt zal het systeem niet werken. Dit voorbeeld is te zien in figuur 3/6.23-13.

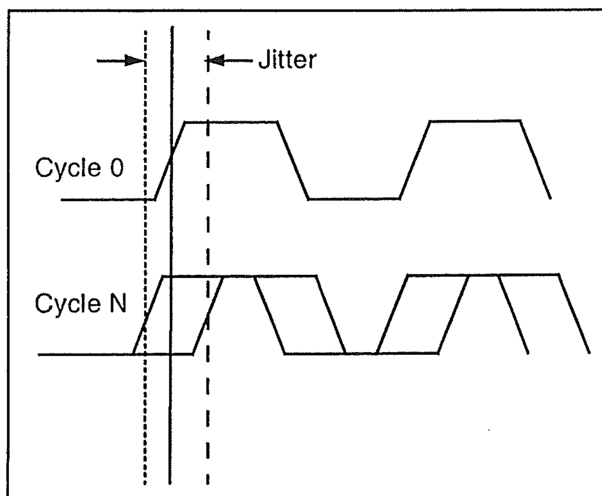


Figuur 3/6.23-12: Periode jitter.

6.23 Werking en principes van clock-drivers/generatoren



Figuur 3/6.23-13: Door periode jitter kan foute data worden opgenomen.



Figuur 3/6.23-14: Lange-termijn jitter.

Lange-termijn jitter

Lange-termijn jitter lijkt veel op periode jitter. Ook hierbij wordt de maximale afwijking van de uitgangstransitie ten opzichte van de ideale positie gemeten, maar dan over "veel" cycli (hoe"veel" hangt af van de toepassing en van de frequentie). Voor PC-moederborden en grafische toepassingen heeft de term "veel" meestal betrekking op 10 tot 20 microseconden. In figuur 3/6.23-14 wordt de lange-termijn jitter grafisch voorgesteld.
Als voorbeeld van een systeem dat door lange-termijn jitter wordt gehinderd wordt als voorbeeld een grafische kaart genomen die een monitor aanstuurt. Veronderstel dat een data-pixel is bedoeld voor de pixel met de coördinaten (10,24) van de beeldbuis. Door de jitter van de clock zou deze data bijvoorbeeld op pixel (11,28) terecht kunnen komen. Aangezien dit effect voor alle pixels bestaat, verschuift het beeld van zijn ideale positie op het scherm. Dit wordt ook wel "weglopen" genoemd.

Oorzaken van jitter

Er zijn vier belangrijke oorzaken van jitter:

- Ruis van de voeding op de ingangen
  Ruis van de voeding op de ingangen van een PLL kan op de uitgangen als jitter verschijnen. Dit is de belangrijkste (hoewel niet altijd constante) oorzaak van jitter. De ruis op de voedingsspanning kan op verschillende manieren tot uiting komen:
  - Ground-bounce
    Wanneer tijdelijk veel stroom door de uitgangsdrivers wordt geleverd ontstaat door de zelfinductie van de bedrading naar de voedingslagen (V_{CC} en GND) een spanningsverlies (waarde = L * di/dt).

Hierdoor wordt het effectieve aardpotentiaal van het component verhoogd. Dus als de uitgangsfrequentie afhankelijk is van de effectieve voedingsspanning, zal deze frequentie door de ground-bounce veranderen. Ten tweede varieert binnen de oscillator ook de drempelspanning van de transistors, waardoor de frequentie verandert. Dit heeft een dubbel effect: ten eerste verandert de uitgangsfrequentie en ten tweede is hier een PLL op aangesloten die de frequentieverandering probeert te corrigeren. Beide effecten verschijnen aan de uitgang als jitter.

- V_{dd}-ruis

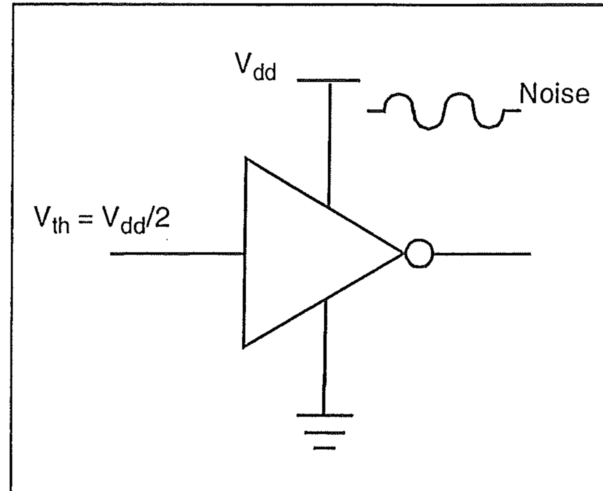
  In figuur 3/6.23-15 is een inverter in de interne teller van de PLL te zien. De drempelspanning van de ingang is de helft van het V_{dd}-potentiaal. Als men aanneemt dat het V_{dd}-signaal een rimpel heeft van 100 mV_{ttt} zal deze een verschuiving in de drempelspanning aan de ingang van de inverter teweeg brengen die op zijn beurt weer jitter veroorzaakt.

  Als dit ruissignaal een stijgtijd van 1 V/ns heeft zal 100 ps piek-piek jitter aan de uitgang van de inverter verschijnen.

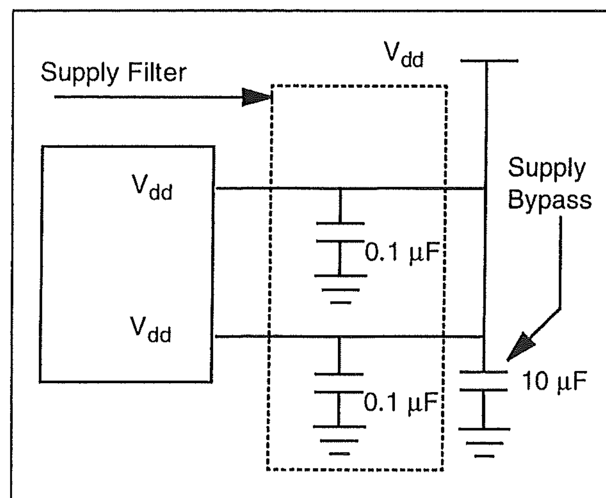
- De PLL in een frequentie-synthesizer heeft een dode band gedurende welke de fase- en frequentie-detector geen kleine veranderingen in de uitgangsfase detecteert.

  Aangezien deze veranderingen niet worden gedetecteerd worden zij ook niet gecorrigeerd en staan zij op de uitgangen in de vorm van jitter.

- Willekeurige thermische ruis van de kristal-referentie of een andere resonerende schakeling.
- Willekeurige mechanische ruis van de kristal-referentie.



Figuur 3/6.23-15: Het effect van V_{dd}-ruis op jitter.



Figuur 3/6.23-16: Een goede ontkoppeling van de voeding zorgt voor een kleinere jitter.

Methoden om jitter te verminderen

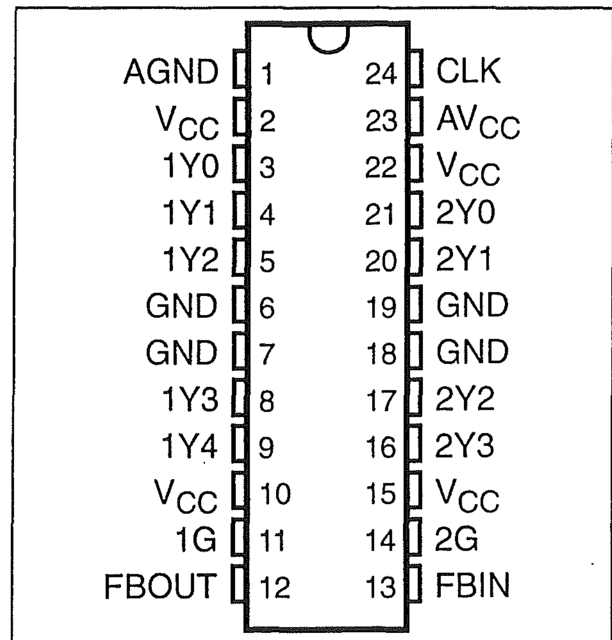
Zoals gezegd zijn ruis op de voedingsspanning en ground-bounce de voornaamste oorzaken van jitter.

Ruis op de voedingsspanning kan worden verminderd door ontkoppeling en filte-

ring (zie ook figuur 3/6.23-16). Groundbounce kan worden verminderd door het aantal belastingen van een uitgang te beperken, door grote aardvlakken op de printkaart aan te brengen en, als er meerdere aardpennen zijn, deze individueel met het aardvlak te verbinden (niet allemaal tegelijk). Ook kunnen de uitgangen van serieweerstanden worden voorzien om de uitgangsstroom te beperken.

Praktisch voorbeeld CDC509

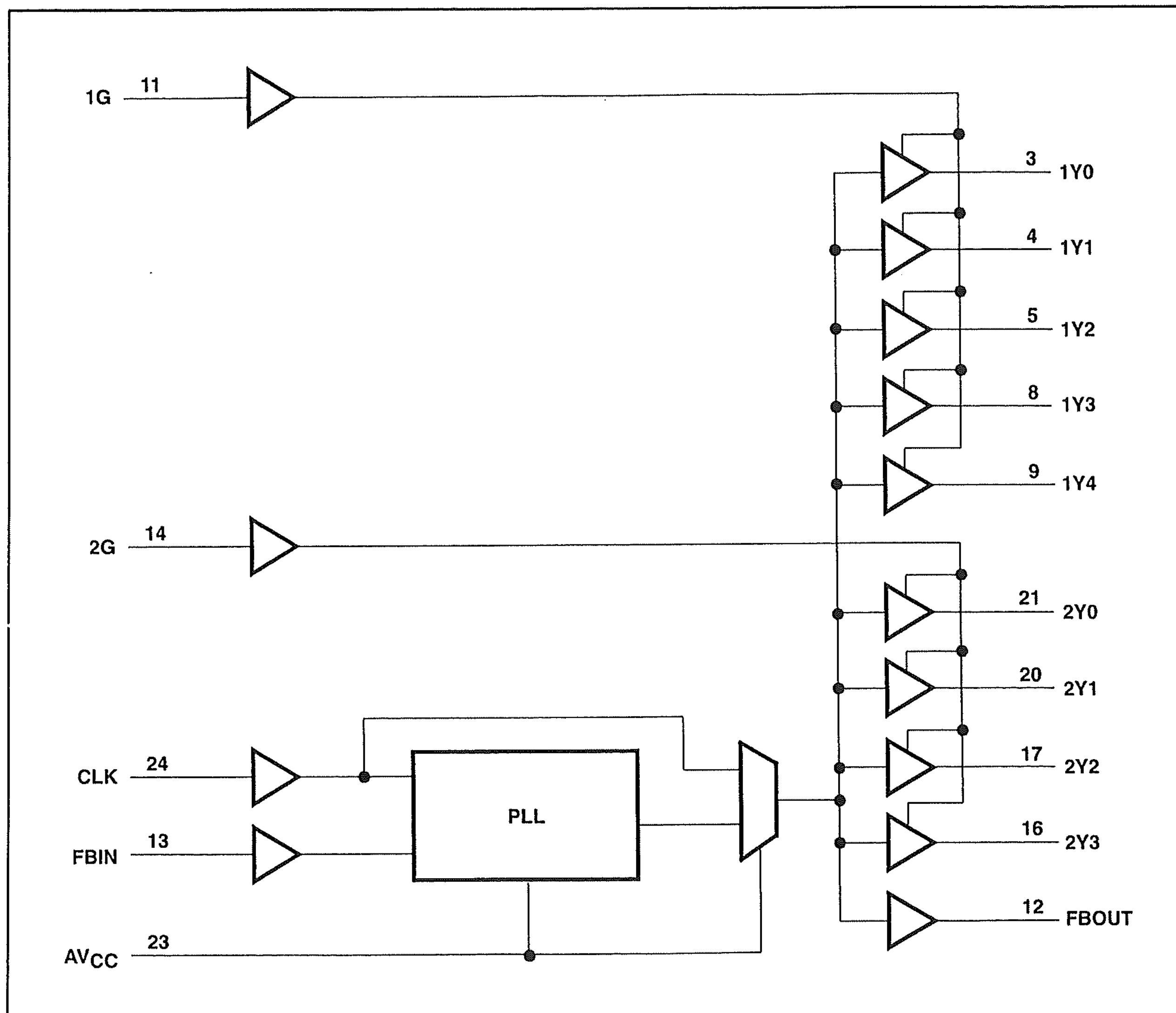
De CDC509 clock-driver
Ten slotte wordt nog een voorbeeld van een praktische, op een PLL gebaseerde, clock-driver beschreven: de CDC509.
De CDC509 is een high-performance phase-lock loop (PLL) clock-driver met kleine skew en weinig jitter. Voor de precieze uitlijning (zowel in frequentie als in fase) van de feedback-uitgang (FBOUT) op het clock-ingangssignaal wordt een PLL toegepast. De CDC509 is speciaal ontworpen voor gebruik met synchrone DRAM's. Hij kan maximaal 5 clock-belastingen per uitgang aansturen en werkt op 3,3 V. Door het gebruik van één bank met 5 uitgangen en één bank met 4 uitgangen ontstaan negen "low-skew", "low-jitter" kopieën van CLK. De duty-cycles van de uitgangssignalen bedragen 50 %, onafhankelijk van de duty-cycle van CLK. Elke bank met uitgangen kan apart worden gesperd of vrijgegeven door middel van de besturingssignalen 1G en 2G. Als de G-ingangen HOOG zijn, schakelen de uitgangen met dezelfde frequentie en fase als CLK. Zijn de G-ingangen LAAG, dan zijn de uitgangen gesperd en bevinden zij zich in de logisch-LAGE toestand. In tegenstelling tot andere componenten die PLL's bevatten, heeft de CDC509 geen externe RC-netwerken nodig. Het loopfilter voor de PLL bevindt zich op de chip. Omdat de CDC509 op een PLL is gebaseerd, is er enige tijd nodig om stabilisatie van de fase-lock van het feedback signaal op het referentiesignaal te verkrijgen. Deze stabilisatietijd is nodig na het opkomen van de voedingsspanning en het aanbrengen van een signaal met een vaste frequentie en fase op de CLK-pen en na eventuele veranderingen van deze signalen. De PLL kan voor testdoeleinden worden omzeild door AV_{CC} aan GND te leggen.



Figuur 3/6.23-17: Aansluitgegevens van de CDC509.

Technische gegevens
- phase-lock loop clock-driver voor synchrone DRAM's
- 1-lijn naar 1 bank van 5 en 1 bank van 4 lijnen
- aparte Output Enable voor elke bank

6.23 Werking en principes van clock-drivers/generatoren



Figuur 3/6.23-18: Intern blokschema van de CDC509.

- geen extern RC-netwerk nodig
- externe feedback-pen (FBIN) voor synchronisatie van de uitgangen op de clock-ingang
- LVTTL-compatibele in- en uitgangen
- voedingsspanning: 3,3 V
- gedistribueerde V_{CC} en GND-pennen
- behuizing: 24-pens Thin Shrink Small Outline (PW)
- fabrikant: Texas Instruments

Aansluitgegevens en intern blokschema
In figuur 3/6.23-17 zijn de aansluitgegevens van deze clock-driver samengevat.

Het intern blokschema is getekend in figuur 3/6.23-18

In de waarheidstabel van figuur 3/6.23-19 wordt de reeds beschreven werking nog eens samengevat.

6.23 Werking en principes van clock-drivers/generatoren

INPUTS			OUTPUTS		
1G	2G	CLK	1Y (0:4)	2Y (0:3)	FBOUT
X	X	L	L	L	L
L	L	H	L	L	H
L	H	H	L	H	H
H	L	H	H	L	H
H	H	H	H	H	H

Figuur 3/6.23-19: Waarheidstabel van de CDC509.

6.23 Werking en principes van clock-drivers/generatoren

3/6.24

Werking en principes van bus-schakelaars

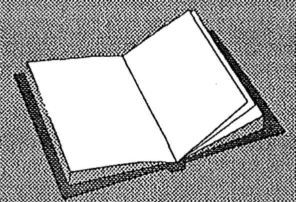
Inleiding

Steeds snellere bussen
Bussen vormen een belangrijke schakel in ieder uitgebreid digitaal systeem, zoals een computer.
Zij verbinden immers alle elektronische blokken van het systeem met elkaar en zorgen voor de primaire data-, adres- en besturings-communicatie tussen deze blokken. In principe zijn alle elektronische blokken van het systeem via bidirectionele schakelaars aangesloten op de bussen. Deze schakelaars zorgen ervoor dat op de juiste momenten de juiste onderdelen via de bussen met elkaar worden verbonden en dat de signalen in beide richtingen door de bussen kunnen stromen.
De snelheden die bussen te verwerken krijgen worden echter steeds hoger. Op moderne PC moederborden wordt met bussnelheden van 100 MHz gewerkt. Hierdoor ontstaan grote problemen bij het schakelen van de bussen. De verschillende blokken van het PC-systeem moeten immers op tijd met de bus verbonden worden, maar ook tijdig van de bus afgeschakeld worden als zij geen gegevens moeten ontvangen of verzenden. Zelfs vertragingen van een paar nano-seconde (ns) kunnen grote problemen veroorzaken.

Weerstanden
Een tweede probleem is dat de onderdelen die de bussen schakelen een zo laag mogelijke inwendige weerstand moeten hebben. Deze inwendige weerstanden staan immers in de weg die de signalen volgen en vormen, samen met de onvermijdelijke paracitaire capaciteiten van het bedradingspatroon van het moederbord, vertragende netwerkjes. Deze netwerkjes kunnen, als de inwendige weerstanden van de bus-schakelaars te hoog zijn, ook vertragingen introduceren die in het ns-bereik liggen.

Bidirectionele busdrivers
Vroeger werden bussen geschakeld via bidirectionele busdrivers uit de TTL-familie, ook wel bus-transceivers genoemd. Deze schakelingen zijn echter niet meer in staat te voldoen aan de eisen die gesteld worden door de moderne snelle buscommunicatie.

LEES OOK:
Hoofdstuk 3/6.23
Hoofdstuk 6/5.8

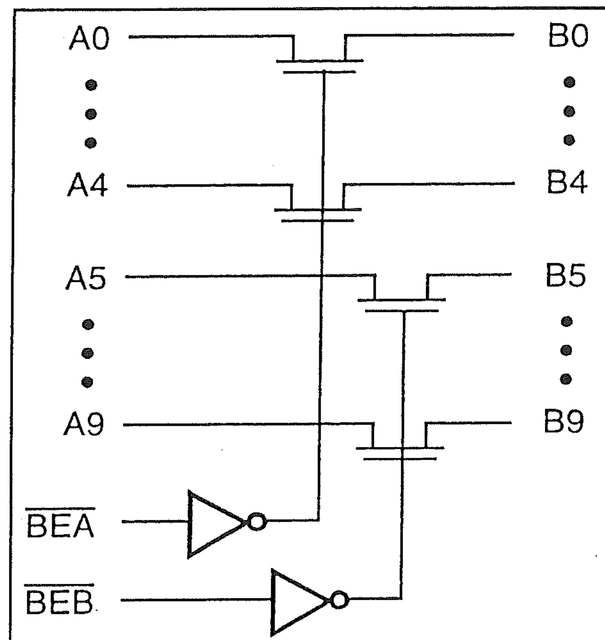


QuickSwitch bus-schakelaars
Tegenwoordig staan voor het schakelen van en naar de bussen echter speciale, uiterst snelle zogenoemde QuickSwitch bus-schakelaars ter beschikking.
Deze bus-schakelaars bieden de volgende voordelen en mogelijkheden:
- zij veroorzaken géén vertraging;
- zij zijn altijd volkomen bidirectioneel;
- conversie van +5 V logica naar +3 V logica is eenvoudig mogelijk;
- er kunnen grote, snelle dual-port RAM's mee worden gemaakt;
- ook "hot-plug" mogelijk.

Eigenschappen
Als voorbeeld worden de QS3383 en QS3384 CMOS bus-schakelaars van Quality Semiconductor behandeld. Als ze enabled zijn, worden twee buslijnen direct met elkaar verbonden via een weerstand van nog geen 5 Ω. Ze gedragen zich als een 5 ns meerpolig relais voor TTL-signalen met een ON-weerstand van 5 Ω. Aangezien deze schakelaars de bus-signalen direct verbinden, introduceren zij geen extra vertraging, timing-skew (zie ook hoofdstuk 3/6.23) of ruis, terwijl zij ook inherent bidirectioneel zijn en geen extra vermogen dissiperen. De bus-schakelaars zijn zodoende zeer geschikt om traditionele TTL-buffers en transceivers te vervangen.

Werkingsprincipe

QS3384
In figuur 3/6.24-1 is het blokschema van de QS3384 CMOS bus-schakelaar te zien. Deze schakeling bestaat uit tien schakelaars, verdeeld over twee banken van 5 stuks. Hierdoor kan de QS3384 worden gebruikt als 10 bit schakelaar of als 5 bit 2-naar-1 multiplexer. De waarheidstabel van deze schakelaar is samengevat in figuur 3/6.24-2.



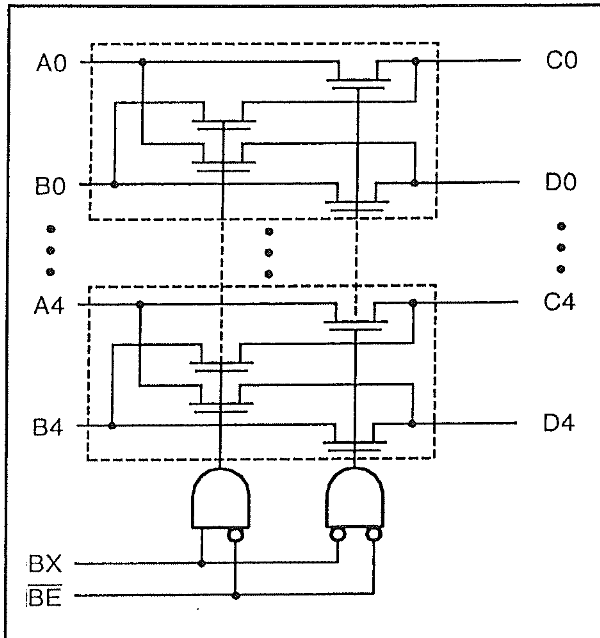
Figuur 3/6.24-1: Het blokschema van de 10-kanaals bus-schakelaar QS3384.

$\overline{BEA}$	$\overline{BEB}$	B4-B0	B9-B5	Function
H	H	Hi-Z	Hi-Z	Disconnect
L	H	A4-A0	Hi-Z	Connect
H	L	Hi-Z	A9-A5	Connect
L	L	A4-A0	A9-A5	Connect

Figuur 3/6.24-2: De waarheidstabel van de QS3384.

QS3383
Het blokschema van een ander type, de QS3383 CMOS bus-exchange schakelaar is getekend in figuur 3/6.24-3.

6.24 Werking en principes van bus-schakelaars

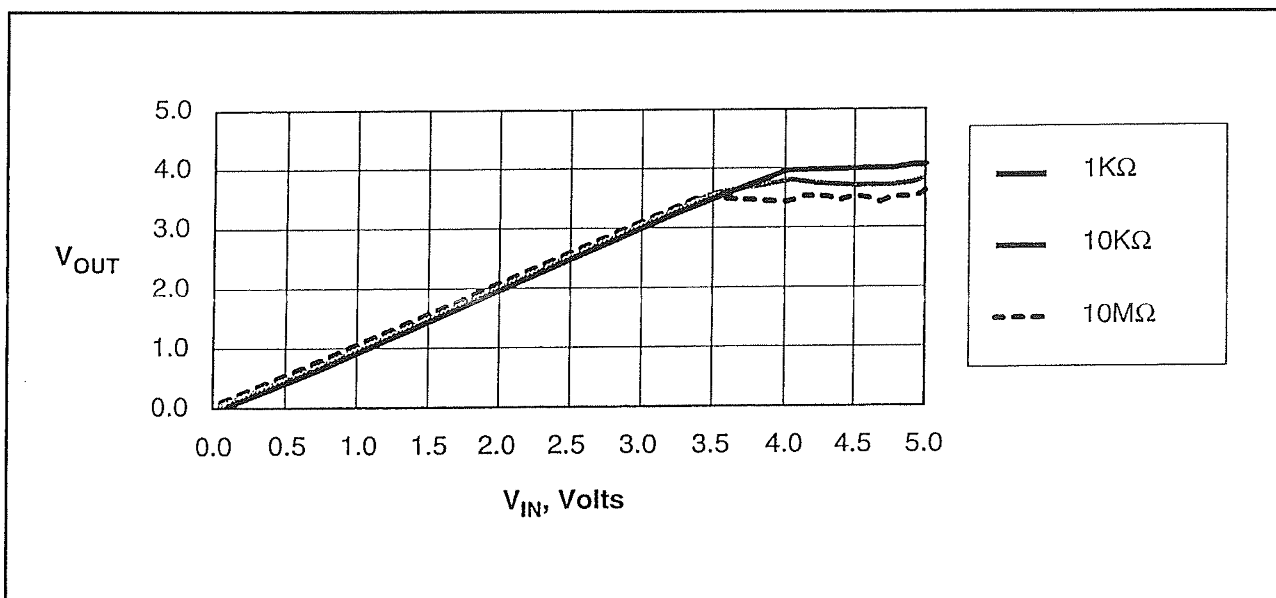


Figuur 3/6.24-3: Het blokschema van de QS3383 bus-exchange schakelaar.

$\overline{BE}$	BX	A4-A0	B4-B0	Function
H	X	Hi-Z	Hi-Z	Disconnect
L	L	C4-C0	D4-D0	Connect
L	H	D4-D0	C4-C0	Exchange

Figuur 3/6.24-4: De waarheidstabel van de QS3383.

Dit IC bestaat uit twee banken van 10 schakelaars die zodanig zijn verbonden dat twee groepen van 5 signalen kunnen worden doorgelaten of verwisseld. Hierdoor kan de QS3383 zowel worden gebruikt als 10 bit schakelaar of als 5 bit twee-richtings bus-wissel schakelaar. De waarheidstabel van deze schakelaar is voorgesteld in figuur 3/6.24-4. De QS3383 is vooral bruikbaar voor wissel- en routing-operaties, zoals byte-swap, crossbar matrices en RAM sharing.

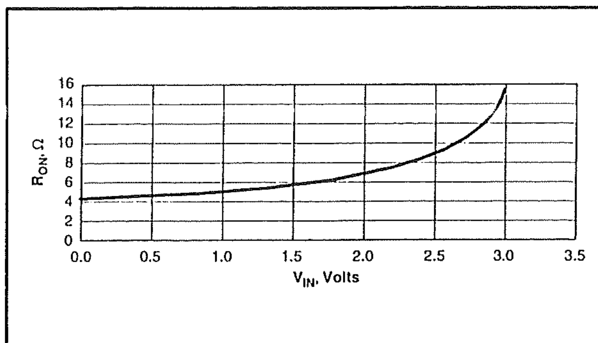


Figuur 3/6.24-5: De uitgangsspanning V_{uit} als functie van de ingangsspanning V_{in} bij verschillende belastingen.

6.24 Werking en principes van bus-schakelaars

Interne opbouw van één schakelaar

Elke schakelaar is opgebouwd uit een N-kanaals MOS transistor die wordt aangestuurd door een CMOS-poort. Als de schakelaar is vrijgegeven (enabled) bevindt de gate van de N-kanaals transistor zich op V_{CC}-niveau (+5 V) en is de schakelaar AAN. Voor lage I/O-spanningen (nabij massapotentiaal) hebben deze schakelingen een ON-weerstand van minder dan 5 Ω en kunnen zij stroomsterkten van ruim 64 mA per kanaal verwerken. Bij hogere I/O-spanningen (bijvoorbeeld TTL-HOOG: +2,4 V) neemt de weerstand iets toe. In dit gebied zijn de A en B pennen (van de QS3384) stevig met elkaar verbonden en wordt de bus-schakelaar op dezelfde manier gespecificeerd als een TTL-schakeling. Wanneer de I/O-spanning tot circa +4 V stijgt, begint de transistor "af te knijpen" (het zogenoemde "pinch-off"-effect), waardoor de voor de belasting beschikbare stroom wordt begrensd, zie figuur 3/6.24-5). Dit komt overeen met een TTL-HOOG toestand van +3,5 V tot +4 V. De ON-weerstand van de schakelaar wordt bepaald door de laagste van de twee spanningen op de I/O-pennen en is eveneens afhankelijk van deze spanningen, zie figuur 3/6.24-6.



Figuur 3/6.24-6: De ON-weerstand R_{ON} van een bus-schakelaar is ook afhankelijk van de ingangsspanning V_{IN}.

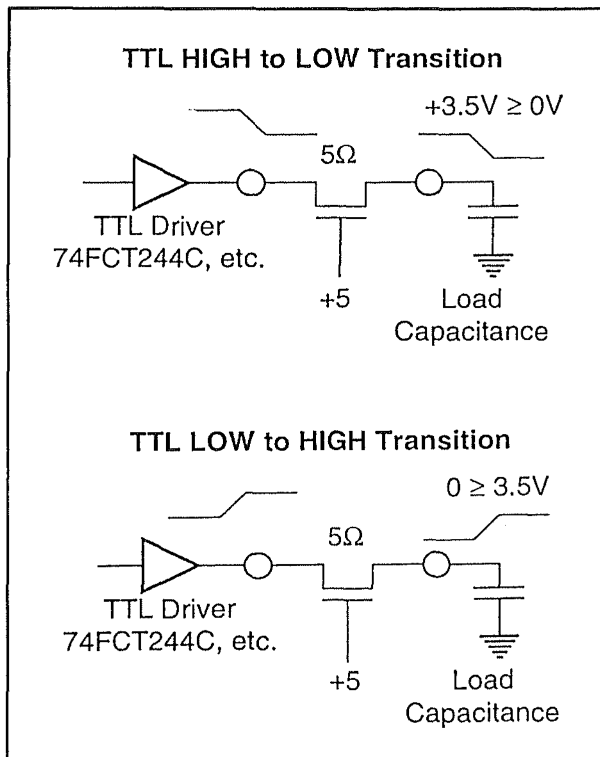
Gedrag bij capacitieve belastingen

De bus-schakelaar voorziet een aandrijvende schakeling van een pad om een capacitieve belasting te laden en te ontladen (zie figuur 3/6.24-7). Als de A (of B) ingang op TTL-LAAG staat (0 V) en de N-kanaals transistor volledig AAN is, zal de B (of A) uitgang deze volgen. Evenzo zal bij een overgang op de A (of B) ingang van TTL-LAAG naar TTL-HOOG niveau de condensator-zijde van de N-kanaals schakelaar zich eerst op 0 V bevinden. De schakelaar is volledig AAN en de B (of A) uitgang zal volgen tot voorbij de drempelspanning. Dit betekent dat de stijg- en afvaltijden (en golfvormen) van de uitgangen worden bepaald door de TTL-driver en niet door de bus-schakelaar. De busschakelaar introduceert dus geen vertraging.
Als de bus-schakelaar gesperd is, bevindt de gate van de N-kanaals transistor zich op 0 V en is de schakelaar UIT. Door de aard van de N-kanaals transistor zijn de pennen A en B dan volledig van elkaar geïsoleerd. Lek en capaciteit gaan dan meer in de richting van de substraat van de chip dan tussen ingang en uitgang. Omdat uitsluitend een N-kanaals transistor wordt gebruikt kan zowel de A- als de B-pen op een niveau van V_{CC} of hoger worden gebracht en kan de voedingsspanning van de component worden verwijderd zonder één van de bussen te belasten.

Vervanging van transceivers

De QuickSwitch bus-schakelaar kan drivers en transceivers in systemen vervangen als het bekrachtigen van de bussen niet nodig is. Aangezien de bus-schakelaar direct twee bussen met elkaar verbindt, levert de schakelaar zélf geen aandrijving, maar is afhankelijk van de schakeling die data op de verbonden bussen zet. Als de

extra belasting van de verbonden bus klein genoeg is, ontstaat er netto een hogere snelheid dan met een driver of transceiver.



Figuur 3/6.24-7: Een CMOS bus-schakelaar als schakelelement tussen een TTL-driver en een capacitieve belasting LOAD.

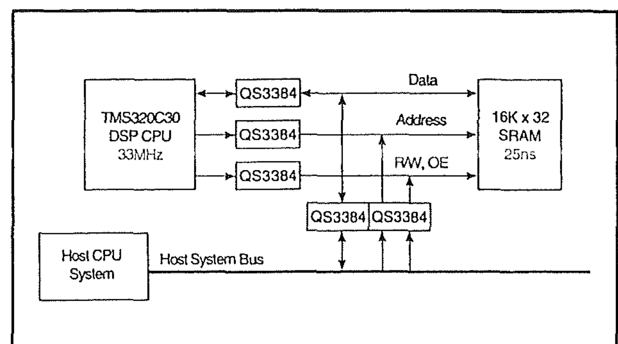
Voorbeeld

De gevoeligheid voor belasting van een driver zoals de 74FCT244 is bijvoorbeeld 2 ns per 100 pF.
Als de verbonden bus 50 pF belasting toevoegt, zal de vertraging hierdoor 1 ns groter worden. Dit is veel minder dan de 4 ns tot 10 ns extra vertraging door de buffer of transceiver die anders gebruikt had moeten worden.

Toepassingsvoorbeelden

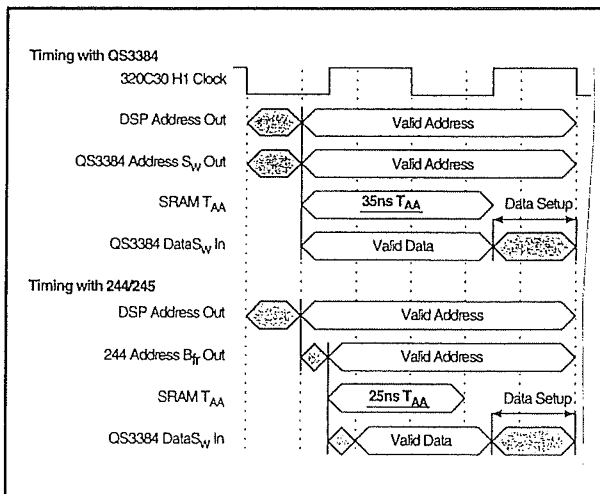
Gedeeld geheugen voor slaaf-processor

Wanneer in een systeem een hoofdprocessor en bijvoorbeeld een signaalprocessor aanwezig zijn, kunnen die beide van hetzelfde geheugen gebruik maken als ook QS3384 bus-schakelaars wordt gebruikt. In figuur 3/6.24-8 is zo'n toepassing getekend van een "host" CPU met een 33 MHz TMS320C30 embedded processor. Beide maken gebruik van dezelfde 16 k x 32 bit SRAM geheugenbank voor de opslag van programma en data. Zowel de CPU als de DSP zijn via enkele QS3384's met de SRAM verbonden. Hiermee wordt 10 ns gewonnen ten opzichte van conventionele buffers en transceivers, namelijk 5 ns voor een 244 adres-buffer naar de SRAM en 5 ns voor een 245 adrestransceiver vanaf de SRAM, zoals uit de timing-diagrammen van figuur 3/6.24-9 duidelijk blijkt. Hierdoor wordt het mogelijk om SRAM's met 35 ns toegangstijd te gebruiken in plaats van de duurdere 25 ns typen.



Figuur 3/6.24-8: Door bus-schakelaars toe te passen kan een slaafprocessor van hetzelfde geheugen gebruik maken als de "host"-processor.

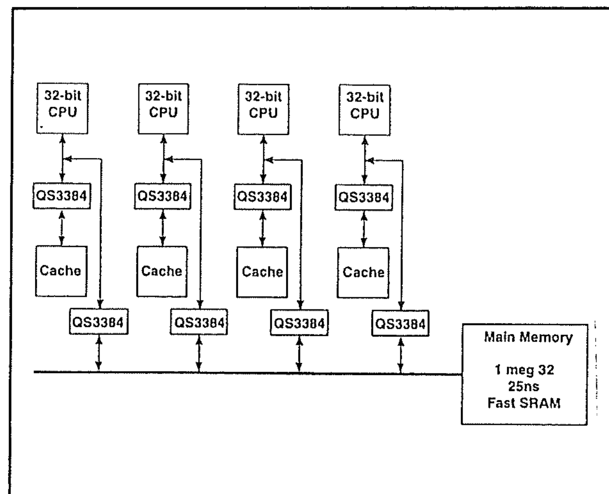
Tussen de berekeningen in wordt de SRAM losgekoppeld van de DSP en aangesloten op de host-CPU waardoor de laatste data kan schrijven vóór de DSP-berekening en data uitlezen na de berekening.



Figuur 3/6.24-9: Door bus-schakelaars toe te passen wordt een tijdwinst geboekt van 10 ns ten opzichte van een 244/245-combinatie. Hierdoor kunnen langzamer (goedkoper) geheugens worden gebruikt.

Multi-processor systeem met snelle bus-verbinding

In figuur 3/6.24-10 is een multi-processor systeem getekend waarbij de QS3384 bus-schakelaar wordt gebruikt als snelle bus-verbinder. In dit voorbeeld heeft elk van de vier processors zijn eigen cache of ander lokaal geheugen, terwijl ze ook gebruik kunnen maken van een 1 M x 32 bit 25 ns SRAM hoofd-geheugen. De QS3384's worden hier gebruikt om adres-, data- en besturingslijnen van de SRAM direct op elke processor aan te sluiten. Wanneer een QS3384 aktief is werkt het hoofd-geheugen als een eenvoudige SRAM samen met de betreffende CPU. Er is dus sprake van een eenvoudige, zeer snelle interface die geen extra vertragingen oplevert.

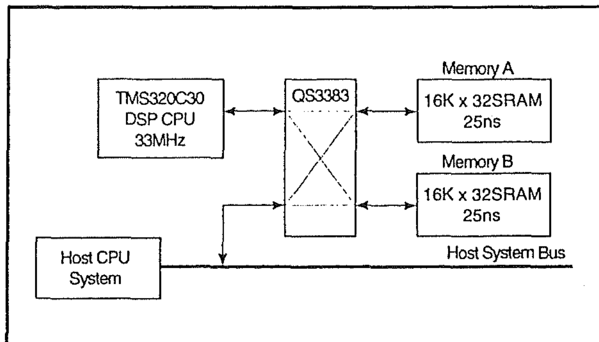


Figuur 3/6.24-10: Een multi-processor systeem waarbij QS3384 bus-schakelaars worden gebruikt.

Bus-exchange schakelaar voor ping-pong geheugen-verbindingen

In figuur 3/6.24-11 is een toepassing getekend van de QS3383 bus-wissel schakelaar. Hierbij maken zowel een DSP-processor als een host-processor gebruik van dezelfde twee geheugens. De QS3383 verbindt geheugen A met de DSP of de host CPU en geheugen B met de host CPU of de DSP, afhankelijk van de toestand van de bus-wissel besturing. Deze configuratie maakt het mogelijk dat de host CPU toegang heeft tot het ene geheugen, terwijl de DSP gebruik maakt van het andere. Als de berekening is beëindigd, worden de geheugens omgewisseld en kan de DSP doorgaan met een volgende berekening terwijl de resultaten van de laatste door de host worden opgehaald.

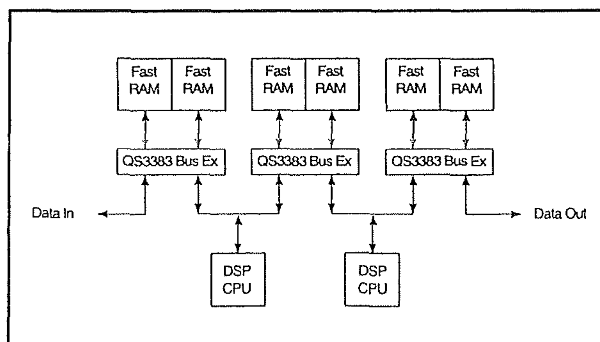
6.24 Werking en principes van bus-schakelaars



Figuur 3/6.24-11: Toepassing van de QS3383 bus-wissel schakelaar als "ping-pong" geheugen-verbinder.

Pijplijnen van processoren

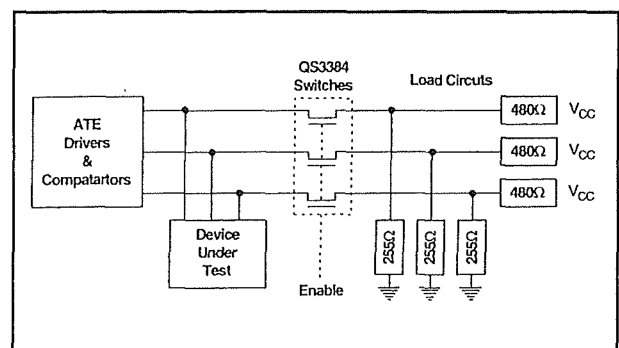
Deze zogenaamde "ping-pong" geheugen-configuratie kan ook worden gebruikt om resultaten tussen verschillende processors te pijplijnen, zie figuur 3/6.24-12. Alle processors brengen data over van hun ingangsgeheugens naar hun uitgangs-geheugens. Wanneer één doorgifte klaar is, worden de geheugens omgewisseld en wordt de uitgang van de ene processor de ingang van de volgende processor.



Figuur 3/6.24-12: Uitbreiding van de schakeling van figuur 3/6.24-11: de "ping-pong" geheugen-verbinders vormen nu een "ping-pong" pijplijn.

ATE belasting-schakelaar

In figuur 3/6.24-13 wordt geïllustreerd hoe de QS3384 kan worden gebruikt als belasting-schakelaar in een automatisch testapparaat (ATE). Voor ATE-toepassingen is telkens een andere opstelling op het belasting-bord nodig (electrical test fixture). Een probleem dat hier vaak bij optreedt is het verbinden en weer losmaken van een belastingsweerstand-netwerk op de te testen aansluitpennen. De belasting moet gedurende gedeelten van de test worden aangesloten en verbroken. Vroeger werd dit gedaan met behulp van kleine relais vanwege de kleine doorlaatweerstand. Relais zijn echter relatief langzaam en groot. De QS3384 kan 10 relais-contacten vervangen, waarbij de schakeltijd wordt verkleind van 5 ms tot 6 ns.

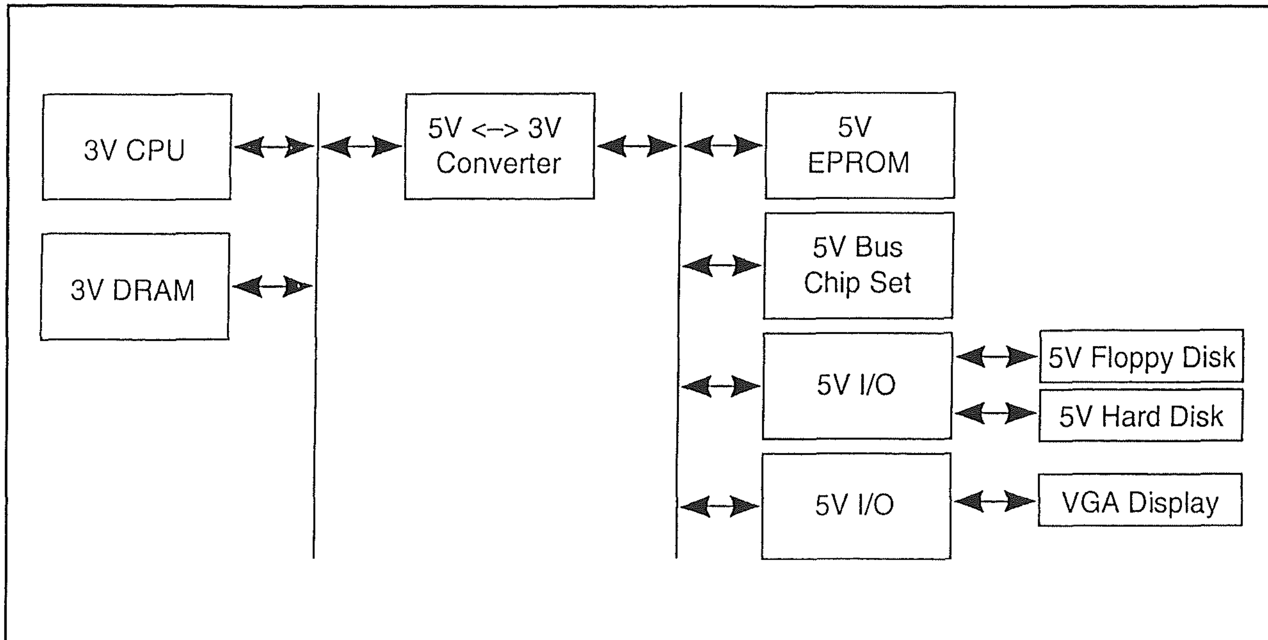


Figuur 3/6.24-13: Toepassing van de QS3384 als belasting-schakelaar in automatische test-apparatuur (ATE).

+5 V naar +3 V omzetting zonder vertraging

Een tegenwoordig veel voorkomend probleem wordt veroorzaakt door het door elkaar gebruiken van +5 V en +3 V logica. Een voorbeeld hiervan is getekend in figuur 3/6.24-14.

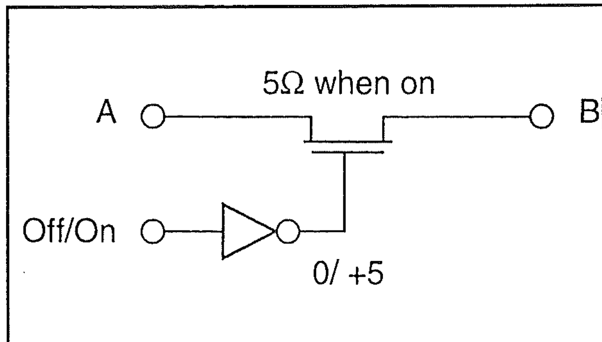
6.24 Werking en principes van bus-schakelaars



Figuur 3/6.24-14: Gemengd gebruik van +5 V en +3 V logica.

De CPU en DRAM hebben een voedingsspanning van +3,3 V, terwijl de EPROM en de I/O-schakelingen op +5 V werken. Tussen beide systemen is een +5 V naar +3 V converter getekend omdat, ook al zijn de logische niveaus gelijk, de +5 V TTL meestal geen +3 V TTL kan aansturen. Een +5 V CMOS-driver zal, bij een logisch HOOG, de bus naar +5 V sturen. De +3 V LVTTL-componenten kunnen niet meer dan ongeveer +3,3 V op hun I/O-pennen verdragen. Wordt toch een hogere spanning op de I/O-pennen aangesloten dan gaat de P-kanaals transistor in de uitgangsdriver geleiden, waardoor er stroom gaat lopen van de bus naar de +3,3 V voeding. Door deze stroom, die zeer groot kan worden, en de bijbehorende latch-up effecten kan de +3,3 V schakeling beschadigd raken.
De functie van de +5 V naar +3 V omzetter is de spanning te begrenzen die gezien wordt door de +3 V logica (typisch niet hoger dan +3,3 V). Een dergelijke omzetter zou kunnen bestaan uit speciale buffers en transceivers die +5 V niveaus aan de ene kant accepteren en +3 V aan de andere kant. Voor deze schakelingen zou echter richtings-besturing en power-sequencing (het in de juiste volgorde inschakelen van de beide voedingen) nodig zijn, terwijl extra vertraging wordt geïntroduceerd.
Een veel betere oplossing is de toepassing van bus-schakelaars zoals de QuickSwitch-familie. Deze leveren bidirectionele +5 V naar +3 V omzetting zonder enige vertraging of richtingsbesturing. Onder de juiste omstandigheden accepteren deze bus-schakelaars +5 V TTL-signalen aan de aandrijvende kant, terwijl de uitgangsspanning wordt begrensd tot +3,3 V aan de aangedreven kant. Bovendien hebben deze bus-schakelaars een ON-weerstand van slechts 5 Ω, zodat geen merkbare vertraging aan de signaalweg wordt toegevoegd. Elke bus-schakelaar bestaat uit een N-kanaals MOS transistor die wordt aangestuurd door een CMOS-schakeling, zie figuur 3/6.24-15.

6.24 Werking en principes van bus-schakelaars

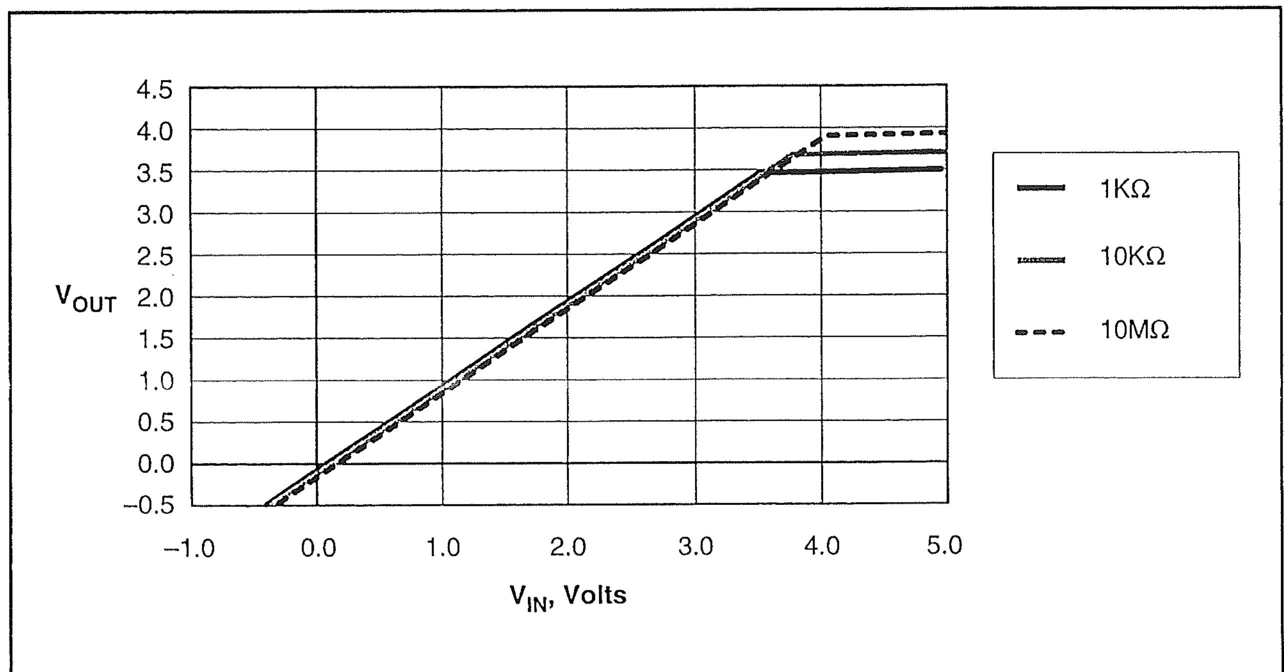


Figuur 3/6.24-15: Een QuickSwitch bus-schakelaar als +5 V naar +3 V omzetter.

Als de bus-schakelaar wordt aangestuurd bevindt de gate van de N-kanaals transistor zich op V_{CC}-niveau (+5 V). De schakelaar is dan AAN en de ON-weerstand bedraagt 5 Ω. Als de bus-schakelaar moet sperren staat de gate op 0 V en is de schakelaar UIT-geschakeld. De lekstroom in deze toestand kan worden omschreven als diode-lek naar de substraat (aarde) en bedraagt meestal iets van 10 nA bij kamertemperatuur.

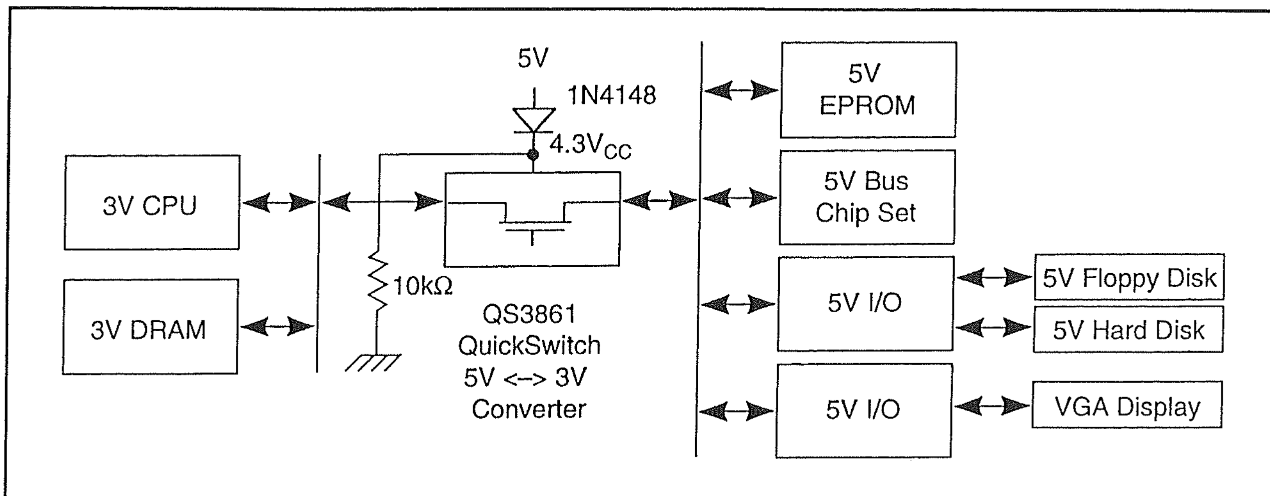
Deze schakelingen hebben een ON-weerstand van minder dan 5 Ω voor ingangsspanningen rond aard-niveau. Deze weerstand neemt iets toe wanneer de I/O-spanningen worden verhoogd van TTL-LAAG naar TTL-HOOG (+2,4 V). In dit gebied zijn de A- en B-pennen stevig met elkaar verbonden. Als de ingangsspanning verder toeneemt, zal de uitgang volgen, maar bij circa +4 V bereikt de uitgangsspanning de hoogste waarde. Bij nog hogere ingangsspanningen zal de uitgangsspanning op +4 V blijven staan. Dit is te zien in de V_{OUT} versus V_{IN}-karakteristiek in figuur 3/6.24-16.

In figuur 3/6.24-17 is getekend hoe het schema van figuur 3/6.24-14 er uit komt te zien als gebruik wordt gemaakt van QuickSwitch schakelaars.



Figuur 3/6.24-16: De uitgangsspanning V_{OUT} van een bus-schakelaar als functie van de ingangsspanning V_{IN}. De uitgangsspanning wordt automatisch begrensd op circa +4 V.

6.24 Werking en principes van bus-schakelaars



Figuur 3/6.24-17: Vervangingsschema van figuur 3/6.24-14 waarbij de plaats van de +5 V naar +3 V omzetter wordt ingenomen door een bus-schakelaar.

3/6.25

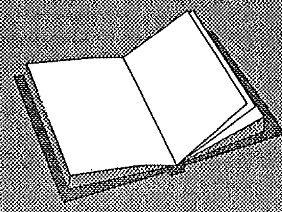
DAS, Data Acquisitie Systemen

Inleiding

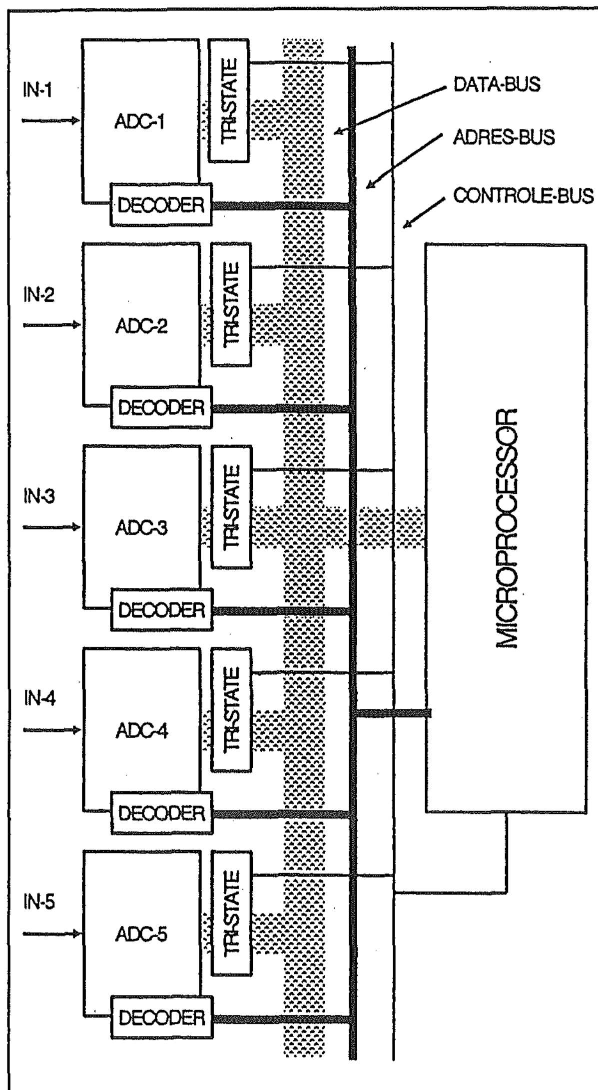
Meer dan één analoog signaal
Analoog naar digitaal omzetters zijn uitstekend geschikt voor het digitaliseren van één analoge ingangsspanning. Deze wordt keurig, op commando van een stuurpuls, omgezet in een binaire code, die verder verwerkt kan worden. Tegenwoordig zal men echter vaak behoefte hebben aan het digitaliseren van meer dan een ingangsspanning. Denk bijvoorbeeld aan industriële processen, waar honderden analoge spanningen en stromen, geleverd door allerlei soorten sensoren, in een computersysteem ingelezen moeten worden. Bij een computergestuurd weerstation zal men minstens een stuk of vijf analoge spanningen moeten verwerken: binnen temperatuur, buiten temperatuur, luchtdruk, luchtvochtigheid, windsnelheid, etc.
Natuurlijk zou men iedere analoge ingangsspanning kunnen aanbieden aan een eigen ADC, maar dan zit men met het probleem dat de digitale uitgangsgegevens op het juiste moment op de een of andere manier op de data-bus van de processor aangeboden moeten worden. De enige oplossing is het blokschema van figuur 3/6.25-1 toe te passen. De digitale uitgangen van de vijf individuele ADC's worden via tri-state buffers op de data-bus van het processorsysteem aangesloten. Deze buffers zijn absoluut noodzakelijk om te verhinderen dat de gegevens van de ene ADC de gegevens van de andere ADC in de weg zitten. Bovendien moet iedere ADC geadresseerd worden. De microprocessor moet weten wanneer de gegevens van een bepaalde ADC op de data-bus staan. Dat is alleen mogelijk als iedere individuele ADC via een adresdecoder op de adres-bus van de processor is aangesloten. In deze adres-decoders wordt aan iedere ADC een individueel adres toegekend, dat softwarematig te benaderen is. Het zal duidelijk zijn dat dit systeem uitstekend werkt, maar nogal wat vraagt van de ontwerper en de printtekenaar!

Een alternatief
Een alternatief is gebruik te maken van maar één ADC, die via een analoge multiplexer verbonden wordt met de verschillende analoge ingangen.

LEES OOK:
Hoofdstuk 6/6.1.7



6.25 DAS, Data Acquisitie Systemen



Figuur 3/6.25-1: Vijf individuele ADC's moeten via vrij uitgebreide hulpschakelingen op de adresbus van een microprocessor worden aangesloten.

Dit systeem wordt voorgesteld in figuur 3/6.25-2 en vraagt aanmerkelijk minder elektronica. De analoge multiplexer wordt gestuurd uit slechts één adresdecoder, rechtstreeks aangesloten op de adres-bus van de microprocessor. De analoge multiplexer zal de vijf analoge ingangen, op commando van de processor, een na een doorverbinden met de ingang van de analoog naar digitaal omzetter. Deze zet de gegevens via slechts één tri-state buffer op de data-bus van het systeem.

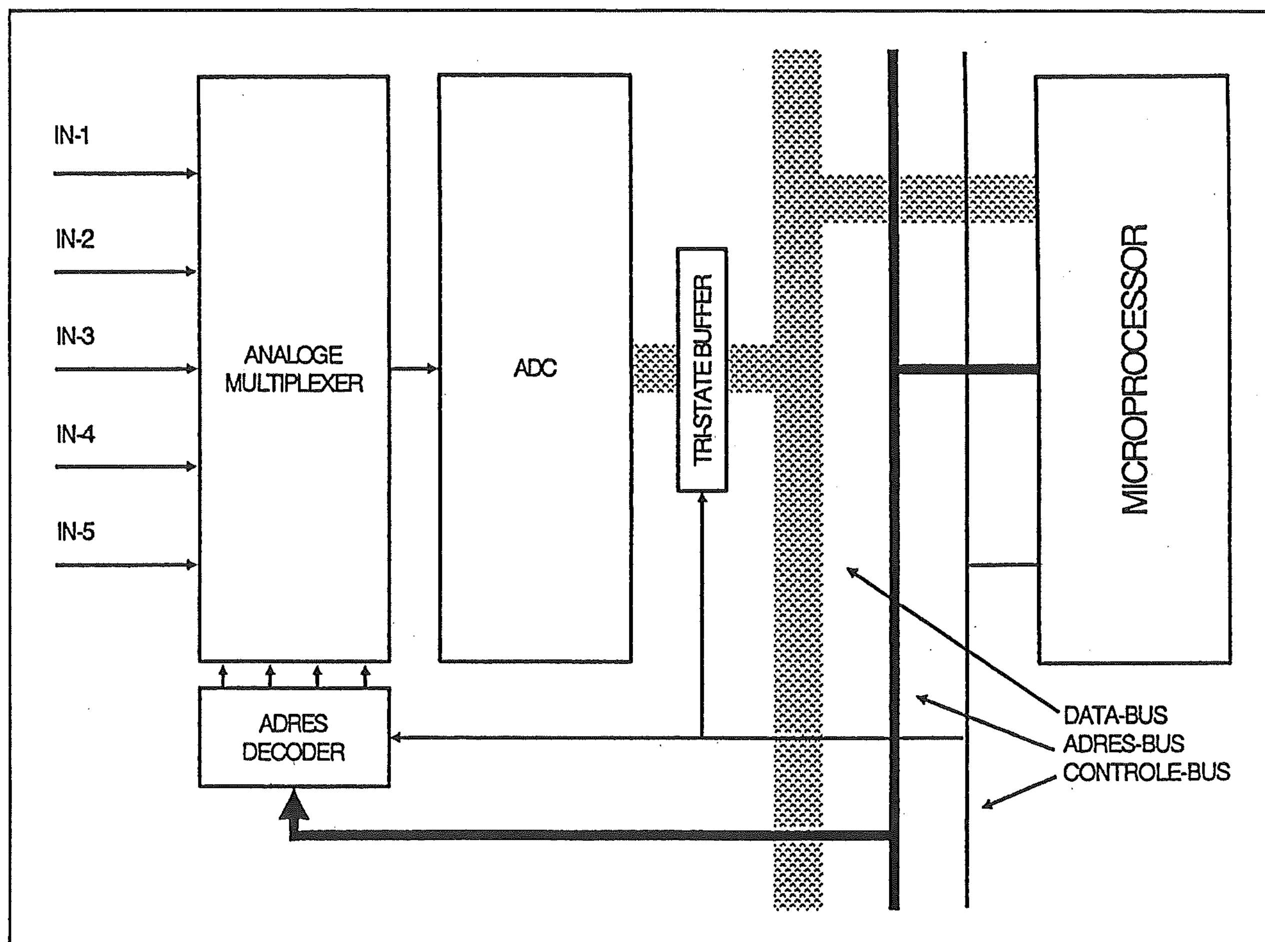
DAS

Het zal duidelijk zijn dat het blokschema van figuur 3/6.25-2 vrij eenvoudig volledig te integreren is. Dat is nu het principe van DAS, afkorting van "Data Acquisition System". Een DAS-IC biedt de mogelijkheid diverse analoge ingangsspanningen op een heel eenvoudige manier een na een om te zetten in een binaire code en deze code's een na een aan te bieden aan de data-bus van een processor-systeem.

Van eenvoudig tot complex

Diverse fabrikanten, waaronder NatSemi, Analog Devices en Plessey, brengen complete DAS-schakelingen op de markt. Er bestaat weinig compatibiliteit tussen de diverse schakelingen. Hoewel het principe van data acquisitie duidelijk is, zijn er natuurlijk tal van opties mogelijk. Zo zijn er vrij eenvoudige IC's, die vier analoge ingangssignalen accepteren en die via een twee bit brede adressering selecteren. Het geselecteerde ingangskanaal wordt rechtstreeks aan een ADC aangeboden en de resultaten van de digitalisatie op een 8 bit brede uitgangsbus aangeboden. Maar daarnaast zijn er ook vrij ingewikkelde schakelingen, die bestuurd moeten worden met een controle-woord dat door een processorsysteem geleverd wordt. In dit controlewoord zit informatie over welke analoge ingang bemonsterd moet worden en hoe dit signaal bemonsterd moet worden, bijvoorbeeld continu of eenmalig. Sommige schakelingen bieden hun digitale informatie rechtstreeks aan op de binaire uitgangen.

6.25 DAS, Data Acquisitie Systemen



Figuur 3/6.25-2: Een alternatieve oplossing, die heel wat minder hardware vraagt.

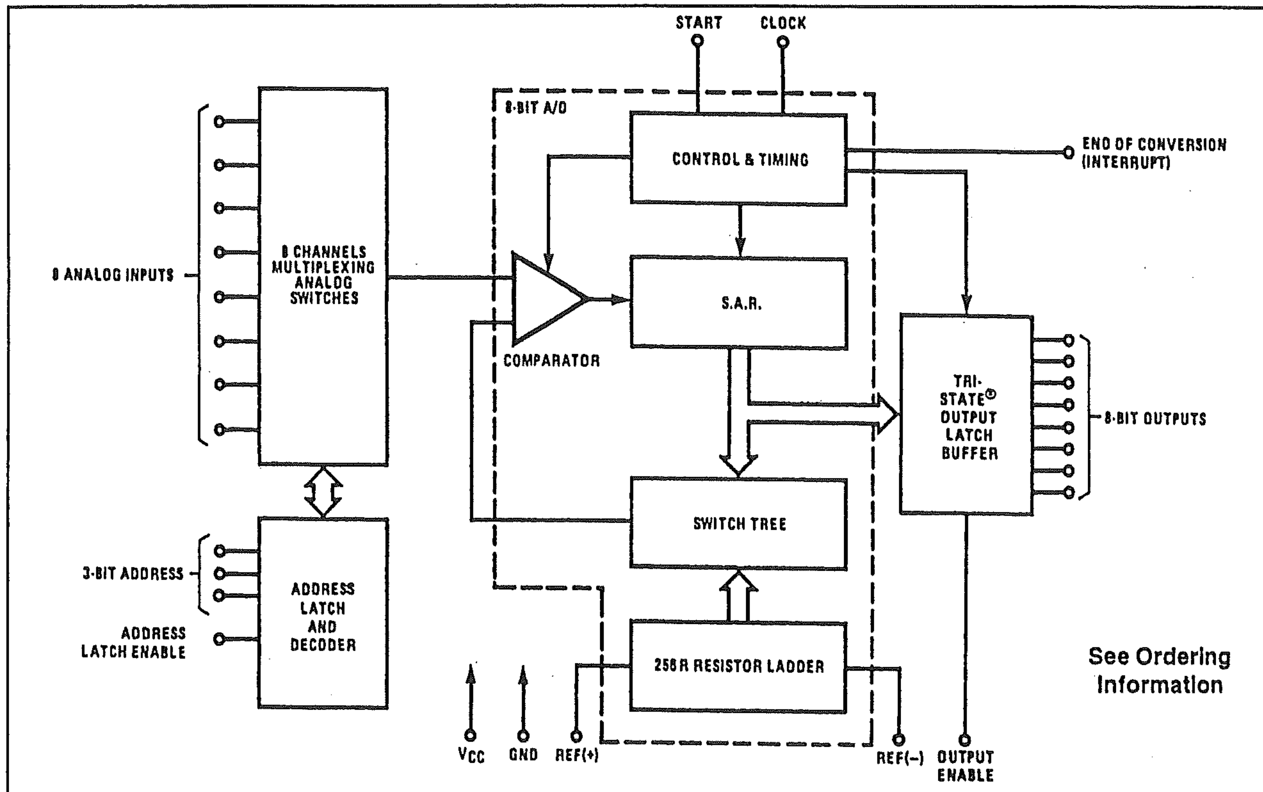
Andere DAS-systemen hebben echter een ingebouwd geheugen, waarin de resultaten van de analoog naar digitaal omzettingen van alle ingangskanalen worden bewaard en op afroep ter beschikking staan. Sommige IC's hebben enkelvoudige ingangen, waarbij de analoge ingangsspanningen worden gerefereerd ten opzichte van de massa. Andere schakelingen bieden echter de mogelijkheid symmetrische spanningen toe te voeren, doordat de ingangstrappen uit twee analoge multiplexers bestaan, waarvan de uitgangen zijn aangesloten op een ingebouwde verschilversterker.
Er is, kortom, voor ieder ontwerpprobleem wel een schakeling te vinden!

Overzicht
In hoofdstuk 6/6.1.7, verschenen in de 69^{e} aanvulling, worden 22 DAS-IC's in het kort voorgesteld.
In dít hoofdstuk wordt als voorbeeld van de werking en de besturing een DAS-IC van NatSemi besproken.

ADC0808: een voorbeeld ter kennismaking

De ADC0808 van NatSemi
Als afsluiting van dit hoofdstuk wordt een typische vertegenwoordiger van de familie der DAS-IC's besproken.

6.25 DAS, Data Acquisitie Systemen



Figuur 3/6.25-3: Het intern blokschema van de ADC 0808.

De ADC0808 is een recht-toe-recht-aan data acquisitie systeem. Zoals uit het intern blokschema van figuur 3/6.25-3 blijkt, bestaat dit IC uit een analoge multiplexer met acht enkelvoudige ingangen. De ingangskanalen worden geselecteerd via de binaire code op drie adresingangen. Dit woord wordt, op commando van het "ADRESS LATCH ENABLE"-signaal, in een latch opgeborgen. De uitgang van de analoge multiplexer gaat naar een standaard ADC die werkt volgens het SAR-principe. Deze SAR wordt uiteraard gestuurd vanuit een clock en de uitgangen van dit register sturen een DAC. De uitgangsspanning van deze DAC wordt in een comparator vergeleken met de analoge spanning die op de uitgang van de analoge multiplexer staat. Als beide spanningen aan elkaar gelijk zijn, wordt de uitgangscode van de SAR in de uitgangslatch opgeborgen. De acht uitgangen van deze latch zijn tri-state uitgevoerd en kunnen dus rechtstreeks op de data-bus van een processor-systeem worden aangesloten.

Controle-ingangen

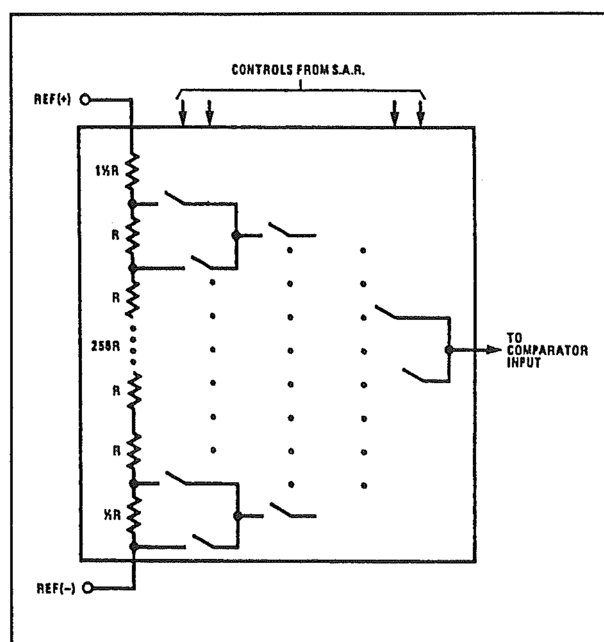
De gehele schakeling heeft maar vier controle-ingangen, namelijk:

- Address Latch Enable (ALE):
  Leest de gegevens op de adres-ingangen in de interne latch in en selecteert een van de acht ingangen.
- Start:
  Geeft het bevel tot het starten van een omzettingscyclus, waarbij de geselecteerde analoge ingangsspanning wordt omgezet in een binaire code.
- Output Enable:
  Stuurt de binaire uitgangen van het IC uit hun tri-state toestand en zet de bi-

6.25 DAS, Data Acquisitie Systemen

naire uitgangsgegevens op de uitgangspennen.

- End of Conversion:
  Geeft een signaal af als de SAR zijn binaire code zo heeft ingesteld, dat de uitgangsspanning van de ingebouwde DAC gelijk is aan de geselecteerde ingangsspanning.



Figuur 3/6.25-4: Het resistieve netwerk in de ADC 0808.

De analoog naar digitaal omzetter

De analoog naar digitaal omzetter in de ADC0808 bestaat uit een resistieve spanningsdeler, een SAR en een comparator. De samenstelling van de resistieve spanningsdeler is getekend in figuur 3/6.25-4. Deze bestaat uit 256 identieke weerstanden en twee weerstanden van respectievelijk 1,5 en 0,5 deze waarde. De totale weerstandswaarde van deze serieschakeling bedraagt typisch 2,5 kΩ. De aftakkingen tussen de weerstanden zijn aangesloten op elektronische schakelaars, die gestuurd worden uit de uitgangen van de SAR. Het schakelaar-netwerk stuurt een van de ingangen van de analoge comparator, de tweede ingang van deze comparator wordt verbonden met de geselecteerde analoge ingang. De SAR zal nu bepaalde schakelaars sluiten, totdat de uitgangsspanning van het schakelaar netwerk gelijk wordt aan de analoge ingangsspanning. Op dat moment staat op de acht binaire uitgangen van de SAR een binair getal, waarvan het binaire gewicht overeen komt met de waarde van de geselecteerde analoge ingangsspanning. In het algemeen kan men stellen dat deze operatie slechts acht clock-pulsen in beslag neemt, zodat na deze tijd het "End of conversion"-signaal gegenereerd kan worden.

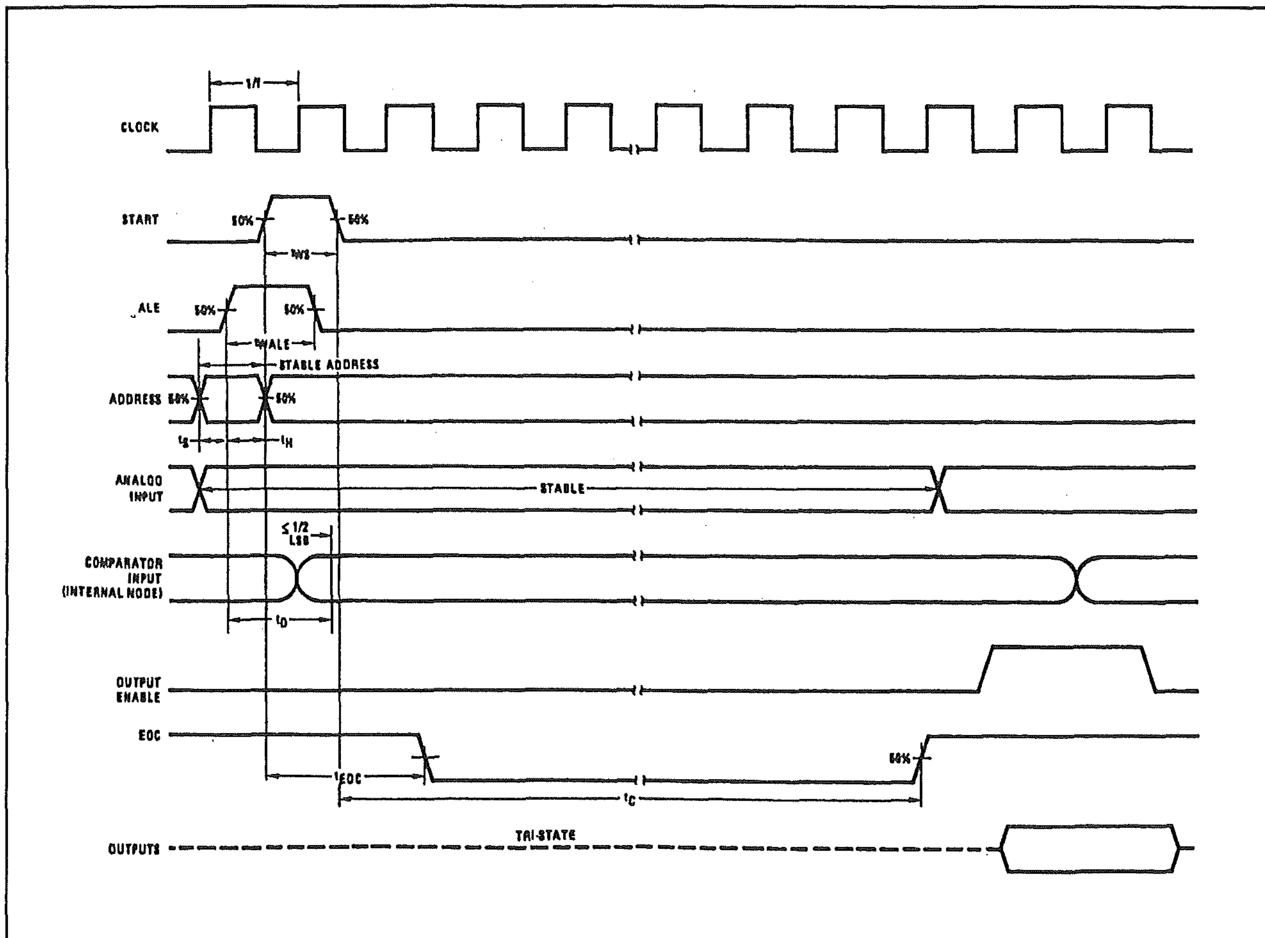
Het selecteren van de analoge ingangen

De acht analoge ingangen worden geselecteerd door een drie bit brede code die op de ADD A, ADD B en ADD C ingangen wordt gezet. De waarheidstabel van deze selectie is weergegeven in figuur 3/6.25-5.

SELECTED ANALOG CHANNEL	ADDRESS LINE C	B	A
IN0	L	L	L
IN1	L	L	H
IN2	L	H	L
IN3	L	H	H
IN4	H	L	L
IN5	H	L	H
IN6	H	H	L
IN7	H	H	H

Figuur 3/6.25-5: De waarheidstabel voor de selectie van de analoge ingangen.

6.25 DAS, Data Acquisitie Systemen



Figuur 3/6.25-6: Het timing-diagram van één omzettingscyclus van de ADC0808.

Het op de ingangen aangemelde adres wordt in de interne latch opgenomen bij de "L" naar "H" overgang van het "ADRESS LATCH ENABLE"-signaal ALE.

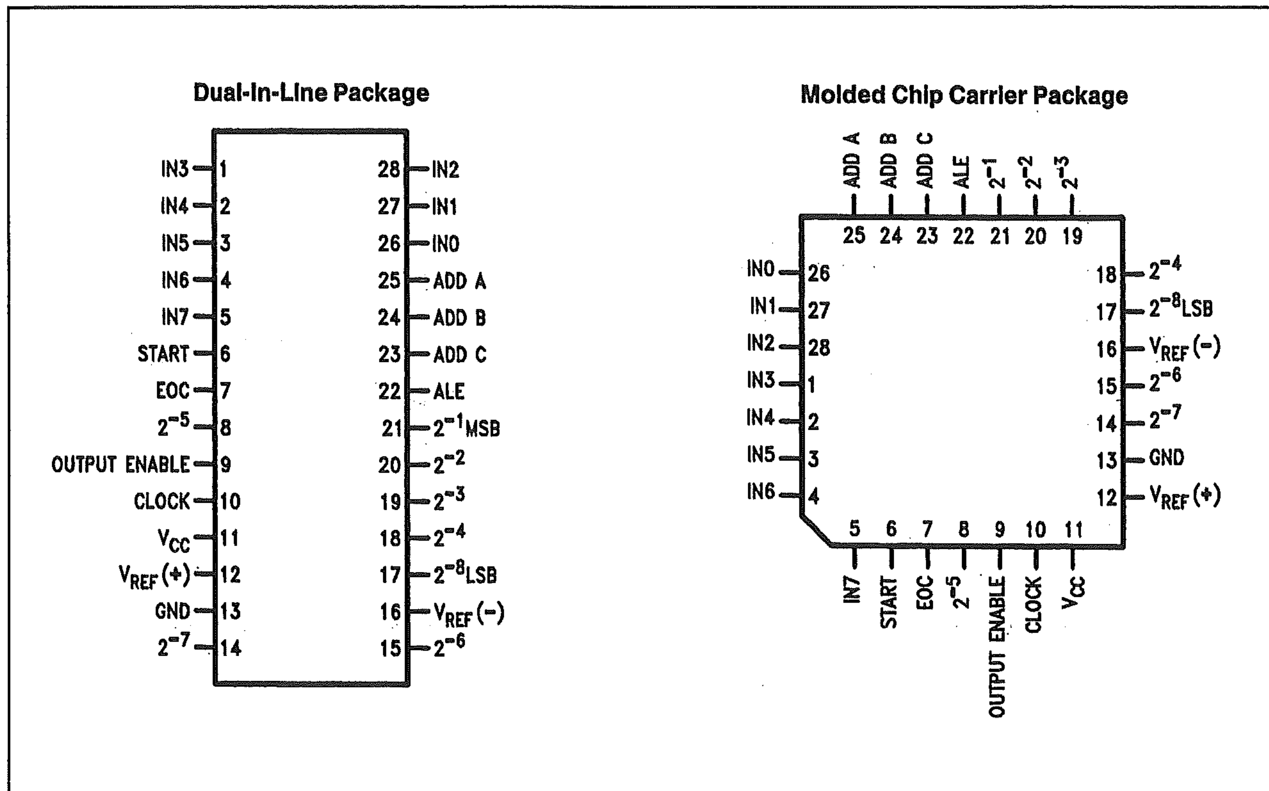
Timing van de schakeling

De volledige timing van de ADC0808 is gegeven in figuur 3/6.25-6. De acht uitgangen van de SAR worden naar "L" gereset op de positieve flank van het "START CONVERSION"-signaal. De omzetting start bij de negatieve flank van dit signaal. Het "END OF CONVERSION"-signaal gaat naar "H" na maximaal acht clockpulsen. Men kan dan een positief "OUTPUT ENABLE"-signaal aanleggen, waarna de acht binaire uitgangen van tri-state naar hun actieve waarde gaan en de resultaten van de omzetting kunnen worden uitgelezen.

Praktische gegevens

De ADC0808 is leverbaar in zowel DIL-28 als in 28-pens chip carrier behuizing. De schakeling wordt gevoed uit een voedingsspanning van +5 V en verbruikt slechts 15 mW vermogen. De weerstandsdeler is uitgevoerd naar de pennen REF_+ en REF_-. De positieve referentie pen kan verbonden worden met een externe referentiespanning, die maximaal gelijk mag zijn aan de voedingsspanning.

6.25 DAS, Data Acquisitie Systemen

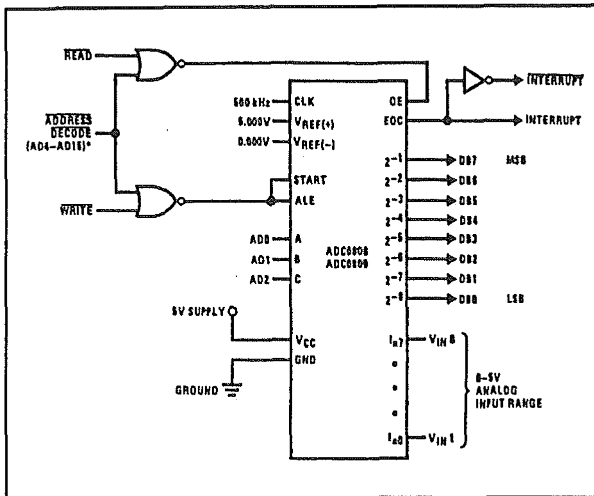


Figuur 3/6.25-7: De aansluitgegevens van de ADC0808.

De negatieve referentie pen ligt meestal aan de massa. Hiermee is al meteen duidelijk, dat de analoge ingangsspanningen kunnen variëren tussen +5 V en 0 V. De comparator vraagt een ingangsstroom van ongeveer 2 μA van de ingangsspanningen, zodat men vaak gebruik zal moeten maken van een buffer als de ingangsspanningen niet belast mogen worden. De clock-frequentie mag maximaal gelijk zijn aan 1,2 MHz, waaruit een minimale conversie tijd van 100 μs afgeleid kan worden. Opgemerkt moet worden, dat de besturingsingangen niet TTL-compatibel zijn. De maximale "L" spanning bedraagt namelijk +1,5 V, bij een voedingsspanning van +5 V en de minimale "H" spanning is gelijk aan +3,5 V. Dat zijn niveaus die echter zonder meer door CMOS-schakelingen geaccepteerd worden.

Voorbeeldschakeling
In figuur 3/6.25-8 is een voorbeeldschakeling rond de ADC0808 getekend, waarbij het IC wordt bestuurd uit een microprocessor. De READ- en WRITE-signalen zorgen respectievelijk voor het uitlezen van de gegevens en voor het starten van een omzetting. Natuurlijk moet de schakeling geadresseerd worden, waarvoor een negatief actief ADRESS DECODE-signaal verantwoordelijk is. Dit signaal kan via een standaard adresdecoder afgeleid worden uit de hoogste bits van de adres-bus. De drie laagste bits moeten natuurlijk vrij blijven, want hiermee moet men een van de acht analoge ingangen selecteren.

6.25 DAS, Data Acquisitie Systemen



Figuur 3/6.25-8: Het besturen van de ADC0808 uit een microprocessorsysteem.

Besluit

De ADC0808 is een overzichtelijke schakeling, die heel eenvoudig te besturen is. Natuurlijk heeft die eenvoud ook bepaalde nadelen. Het grootste nadeel is dat de gegevens niet in een intern geheugen worden opgeborgen, zodat de ontwerper/ster zélf extra schakelingen moet verzinnen (bijvoorbeeld latches) of de noodzakelijke software moet schrijven om het geheugen van een computer aan te spreken.

3/6.26

Werking en principes van monostabiele multivibratoren

Definitie
Een monostabiele multivibrator is een schakeling die een uitgangspuls met instelbare tijdsduur afgeeft, onafhankelijk van het triggersignaal aan de ingang.

74121 als voorbeeld
In dit hoofdstuk wordt aan de hand van het nog steeds veel gebruikte type 74121 uitgelegd hoe een monostabiele multivibrator (ook wel one-shot of MMV genoemd) werkt en waar hij voor gebruikt kan worden.

Werking
In figuur 3/6.26-1 is het blokschema van de 74121 getekend en in figuur 3/6.26-2 het hiermee overeenkomstige logische symbool volgens de IEC-norm. Figuur 3/6.26-3 is de bijbehorende waarheidstabel. Zoals in de onderste vijf rijen van de waarheidstabel te zien is kan de one-shot worden getriggerd door middel van twee negatief-actieve ingangen A1 of A2 (met ingang B = "H") of door een positief-actieve ingang B (met A1 en/of A2 = "L"). Het triggeren (starten van de uitgangspuls) gebeurt op een bepaald spanningsniveau aan de ingang en is niet direct afhankelijk van de stijg- of daaltijd van de ingangspuls. De B-ingang is uitgevoerd als echte Schmitt-trigger met hysteresis, waardoor ook langzame niveau-veranderingen (tot 1 V/s) een exacte triggering opleveren.

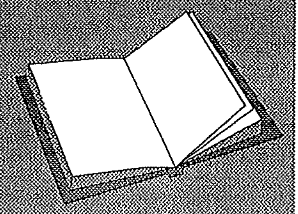
De ingangspulsen mogen elke lengte ten opzichte van de uitgangspuls hebben. Zolang de puls aan de uitgang aanwezig is, hebben veranderingen aan de ingangen geen enkele invloed meer en is de tijdsduur van de uitgangspuls alleen afhankelijk van de tijdsbepalende componenten. Wanneer geen externe timingcomponenten worden aangesloten (R_{int} aan Vcc met C_{ext} en R_{ext}/C_{ext} open) duurt de uitgangspuls 30 ns tot 35 ns.
De externe condensator mag variëren van 10 pF tot 1.000 μF en de externe weerstand van 1,4 kΩ tot 40 kΩ.
De tijdsduur van de uitgangspuls wordt bepaald door:

$$t_{w(out)} = C_{ext} * R_T * \ln 2$$

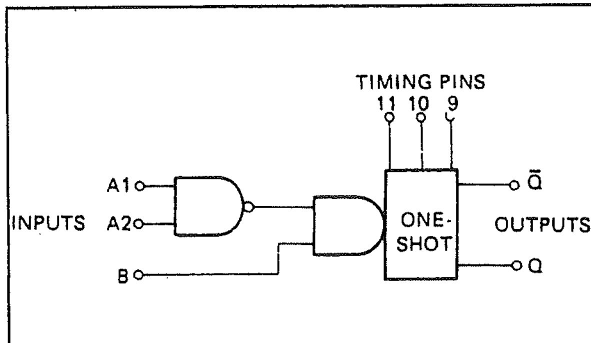
$$t_{w(out)} = 0,7 * C_{ext} * R_T$$

In de figuren 3/6.26-4 en -5 zijn voorbeelden van de optredende golfvormen getekend.

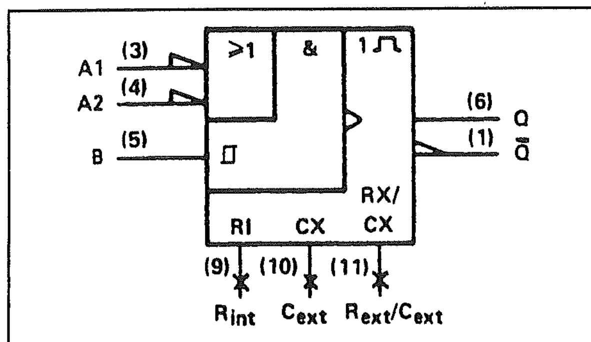
LEES OOK:
Hoofdstuk 3/6.6
Hoofdstuk 3/99.6



6.26 Werking en principes van monostabiele multivibratoren



Figuur 3/6.26-1: Blokschema van de 74121.



Figuur 3/6.26-2: IEC-symbool van de monostabiele multivibrator 74121.

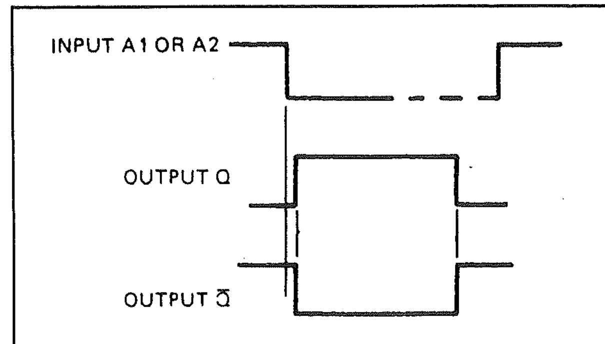
FUNCTION TABLE

INPUTS			OUTPUTS	
A1	A2	B	Q	$\bar{Q}$
L	X	H	L	H
X	L	H	L	H
X	X	L	L	H
H	H	X	L	H
H	↓	H	⊓	⊔
↓	H	H	⊓	⊔
↓	↓	H	⊓	⊔
L	X	↑	⊓	⊔
X	L	↑	⊓	⊔

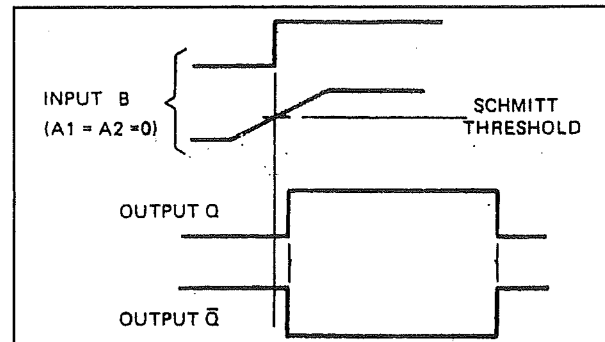
Figuur 3/6.26-3: De waarheidstabel van de 74121.

Opmerking

Bij een hertriggerbare MMV kan de uitgangspuls wel opnieuw worden gestart terwijl deze nog niet afgelopen is. De pulsduur wordt zodoende verlengd.



Figuur 3/6.26-4: Golfvormen bij triggeren op de A-ingang.



Figuur 3/6.26-5: Golfvormen bij (eventueel vertraagd) triggeren op de B-ingang.

One-shot met vertraagde ingang

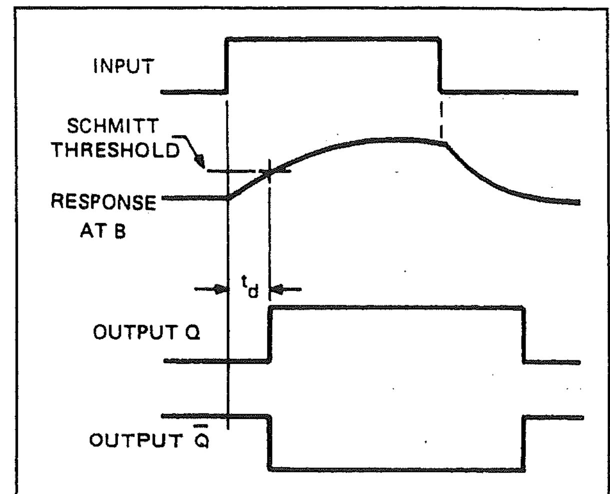
De B-ingang is door zijn positieve flankgetriggerde Schmitt-trigger eigenschap ideaal geschikt om het ingangssignaal vertraagd te ontvangen via een RC-netwerk. De schakeling is getekend in figuur 3/6.26-6, de timing in figuur 3/6.26-7. Als de ingang verandert van "L" naar "H" wordt de condensator C in de richting van "H" opgeladen. Na een tijd t_d heeft de spanning over de condensator het Schmitt-trigger-niveau bereikt, waardoor de monostabiele multivibrator "ontsteekt" en de uitgang een puls afgeeft. Om te voorkomen dat de one-shot een puls afgeeft bij een "L" ingangssignaal, mag de waarde van R niet te groot zijn. Aan de hand van de figuren 3/6.26-8 en -9 kan

6.26 Werking en principes van monostabiele multivibratoren

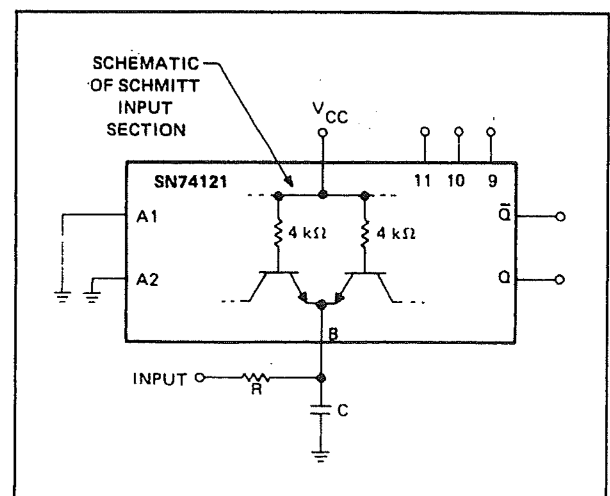
men de maximale waarde van R uitrekenen. De spanning V_B op punt B mag, om herkend te worden als logische "L", niet hoger zijn dan 0,8 V. Bovendien is:

$V_B = [(V_{cc} - V_F) * R + V * V_iR_i)] / (R + R_i)$

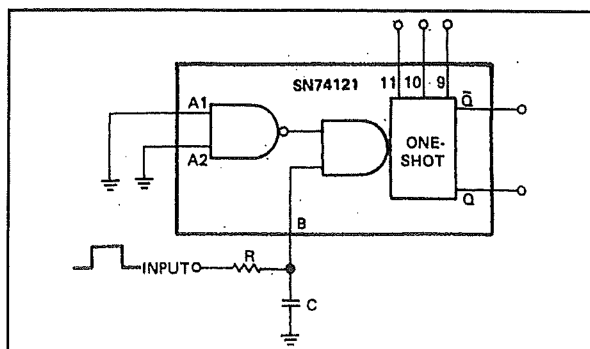
Uitgewerkt geeft dit 427 Ω als maximaal toelaatbare waarde van R. De maximale waarde van C wordt bepaald door de "duty-cycle" (aan/uit-verhouding) van de golfvorm aan de ingang en de waarde van R. C moet immers zijn ontladen voordat de volgende cyclus kan beginnen. De vertragingstijd t_d is recht evenredig met de waarde van condensator C en (R + r). Hierin is r de uitgangsimpedantie van de sturende trap in de "H"-toestand (meestal zo'n 130 Ω). Er bestaat geen eenvoudig verband tussen R en de vertragingstijd vanwege de Schmitt-ingangsstroom die door R vloeit. In de tabel van figuur 3/6.26-10 zijn enkele voorbeelden van vertragingstijden met R = 100 Ω en variërende C te zien.



Figuur 3/6.26-7: De golfvormen van figuur 3/6.26-6.



Figuur 3/6.26-8: Opengewerkte B-ingang van de 74121.



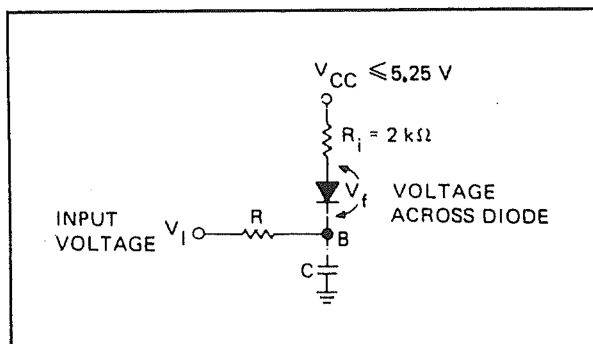
Figuur 3/6.26-6: Het vertraagd triggeren door middel van een RC-netwerk op de B-ingang.

Stabiele oscillator

Met behulp van twee MMV's kan een "clock-generator" worden samengesteld die de golfvormen van figuur 3/6.26-12 genereert. Het schema is getekend in figuur 3/6.26-11.

Deze oscillator kan op twee manieren tot werken worden gedwongen, door Gate 1 LAAG te maken of Gate 2 HOOG. In figuur 3/6.26-13 zijn de uitgangssignalen van de schakeling getekend. De tijden t1 en t2 worden bepaald door de tijdconstanten R1*C1, respectievelijk R2*C2. Deze schakeling kan bijvoorbeeld praktisch worden toegepast als goedkope zelfbouw pulsgenerator met variabele frequentie.

6.26 Werking en principes van monostabiele multivibratoren

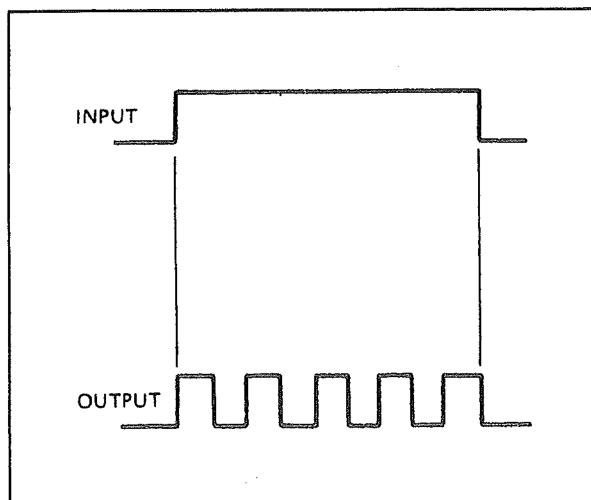


Figuur 3/6.26-9: Vereenvoudigde voorstelling van figuur 3/6.26-8.

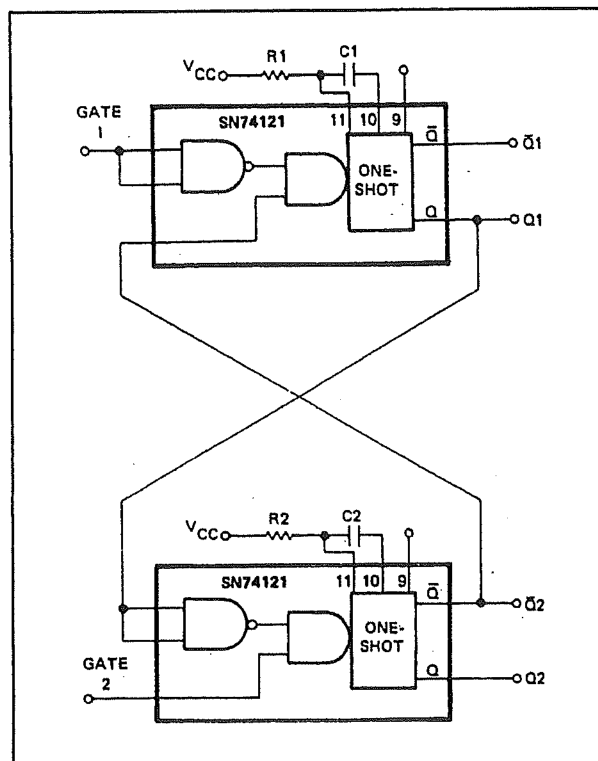
gemeten vertragingstijden bij verschillende condensator-waarden

t_d (µs)	C(nF)
1.36	10
2.50	20
4.90	40
12.46	1.12

Figuur 3/6.26-10: Vertragingstijden bij verschillende RC-tijdconstanten.

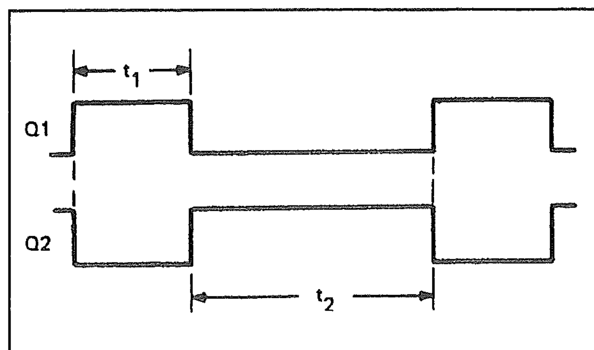


Figuur 3/6.26-12: Verband tussen de spanning op G2 (boven) en de spanning op de uitgang (onder).



Figuur 3/6.26-11: Eenvoudige blokgolfgenerator met twee MMV's.

Moet de uitgangspuls hiervan dan ook variabel zijn (zonder de frequentie te beïnvloeden), dan dient nog een extra MMV aan een van de uitgangen te worden geplaatst.



Figuur 3/6.26-13: De uitgangsspanningen op Q1 en Q2.

3/6.27

Werking en principes van FIFO's

Principes

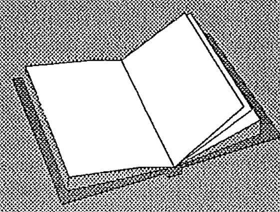
Inleiding

Het komt vaak voor dat verschillende digitale systemen met verschillende snelheden werken. In feite is het veiliger ervan uit te gaan dat dit altijd het geval is. Microprocessoren en bijbehorende geheugens werken bijvoorbeeld veel sneller dan andere componenten, zoals het toetsenbord en de floppy disk.

Wanneer informatie van het ene subsysteem naar het andere moet worden verplaatst, wordt de snelheid daarvan beperkt door het langzaamste systeem. Het snelle systeem moet telkens even wachten totdat het langzame de data heeft overgenomen. Wanneer dat niet mogelijk of niet gewenst is, moet het snelheidsverschil worden overwonnen door een tussenliggende databuffer. Bij parallel transport moet de buffer het aantal bits per woord kunnen opnemen. Bovendien moet hij lang genoeg zijn om alle woorden die in een bepaalde periode moeten worden getransporteerd (een data-blok) te kunnen bevatten. Wanneer de data die het eerst in de buffer gaat er aan de andere kant ook weer als eerste uit komt, spreekt men van een FIFO (First-In First-Out) geheugen of register. De data schuift daarbij steeds zo ver mogelijk door naar achteren. Het eerste woord komt dus helemaal achteraan, het tweede woord komt één plaats minder ver terecht, enzovoort. De FIFO werkt asynchroon als het laden aan de ene kant geheel onafhankelijk van het lossen aan de andere kant kan gebeuren.

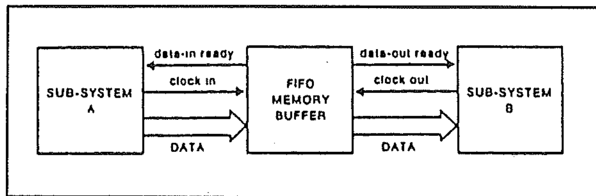
Het is natuurlijk wel handig als de verzendende kant weet of er nog plaats is in de FIFO. Aan de ingang is dan ook een besturingssignaal aanwezig dat dit aangeeft: "Data-In Ready" (DIR) is meestal HOOG als er nog plaats is. Als de FIFO vol is, gaat dit signaal LAAG en blijft LAAG totdat er data van de uitgang wordt afgenomen. De laatste plaats komt dan vrij en alle overblijvende data schuift dus tegelijk één plaats naar achteren op en DIR wordt weer HOOG. Hetzelfde geldt natuurlijk ook voor de ontvangende kant. Aan de uitgang van de FIFO is een besturingssignaal beschikbaar dat aangeeft of de buffer nog data bevat: "Data-Out Ready" (DOR) is meestal HOOG als er nog data aanwezig is. Is de FIFO leeg dan gaat DOR LAAG

LEES OOK:
Hoofdstuk 3/6.10
Hoofdstuk 3/6.13



6.27 Werking en principes van FIFO's

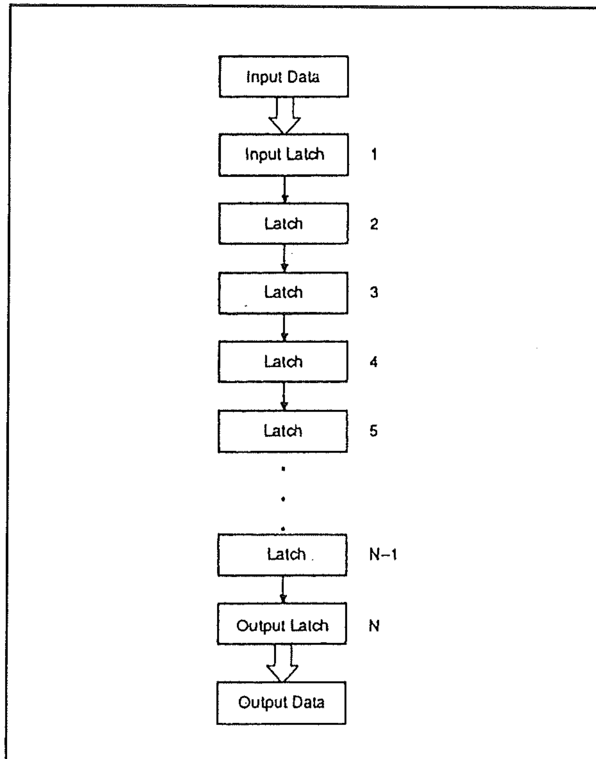
en blijft LAAG totdat er weer data (aan de uitgang) aanwezig is.
Het clocksignaal waarmee de data wordt ingeschreven wordt Shift-In, Clock-In of Load Clock genoemd. De data wordt uitgelezen met Shift-Out, Clock-Out of Unload Clock, zie figuur 3/6.27-1.



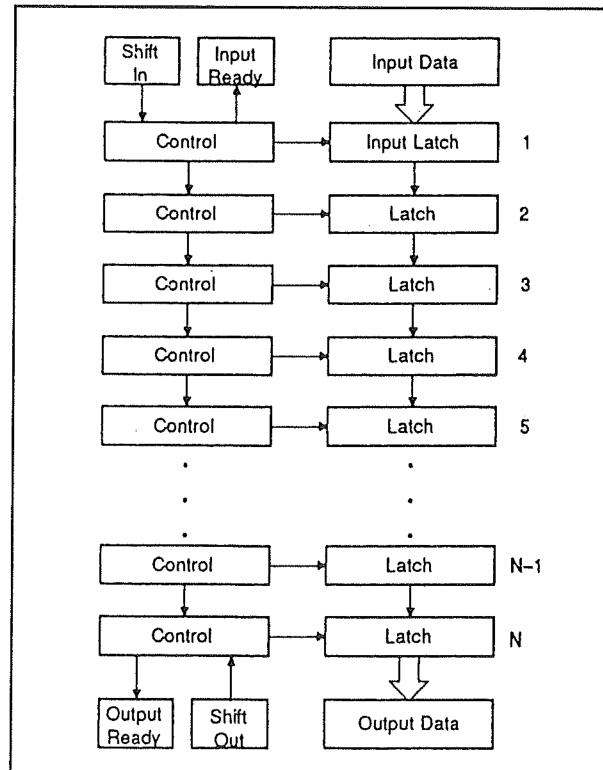
Figuur 3/6.27-1: Bij datatransport tussen twee systemen moet het verschil in snelheid worden opgevangen door een asynchrone buffer.

FIFO principes

Voor het transporteren van de data binnen de FIFO is geen clock nodig. Van buitenaf gezien schuift de data vanzelf door naar achter. Het spreekt vanzelf dat FIFO's op verschillende manieren samengesteld kunnen worden. De eerste gedachte is uit te gaan van een schuifregister of een aantal achter elkaar geplaatste latches. In deze vorm "valt" de data echter niet door naar achter, maar schuiven alle woorden telkens één plaats tegelijk op. Is zo'n register bijvoorbeeld N woorden lang, dan zijn N klokpulsen nodig om de data weer aan de uitgang te laten verschijnen: de FIFO heeft een vaste lengte, zie figuur 3/6.27-2.

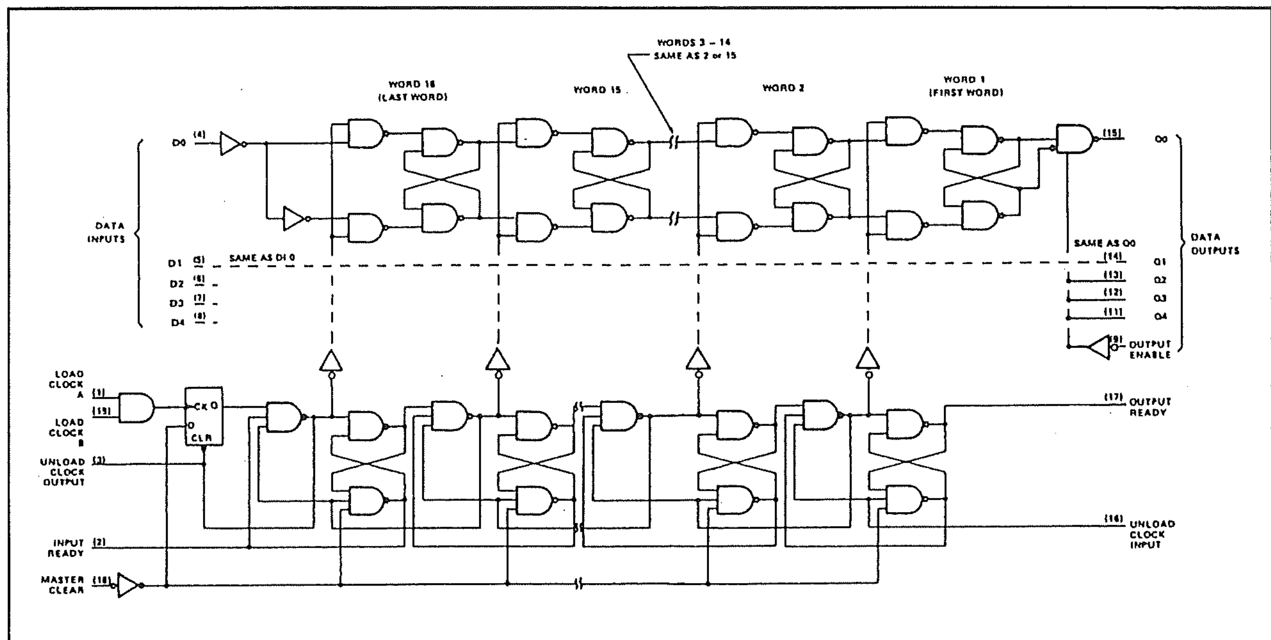


Figuur 3/6.27-2: Opbouw van een FIFO met vaste lengte door middel van achter elkaar geplaatste latches.

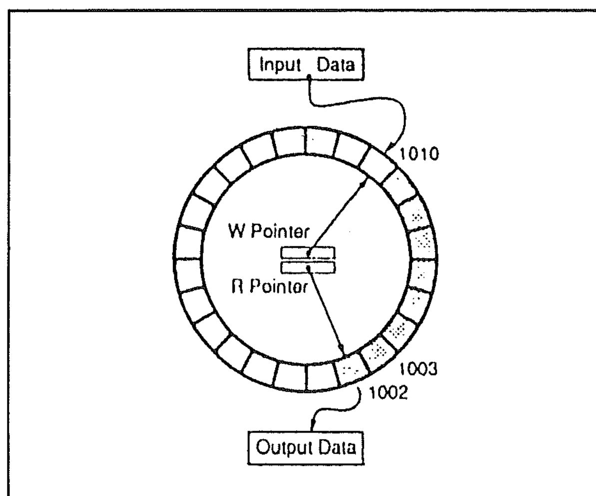


Figuur 3/6.27-3: Principiële opbouw van een FIFO met variabele lengte.

6.27 Werking en principes van FIFO's



Figuur 3/6.27-4: Functioneel blokschema van een uit latches en flip-flop's samengestelde FIFO met variabele lengte (type 74S225).

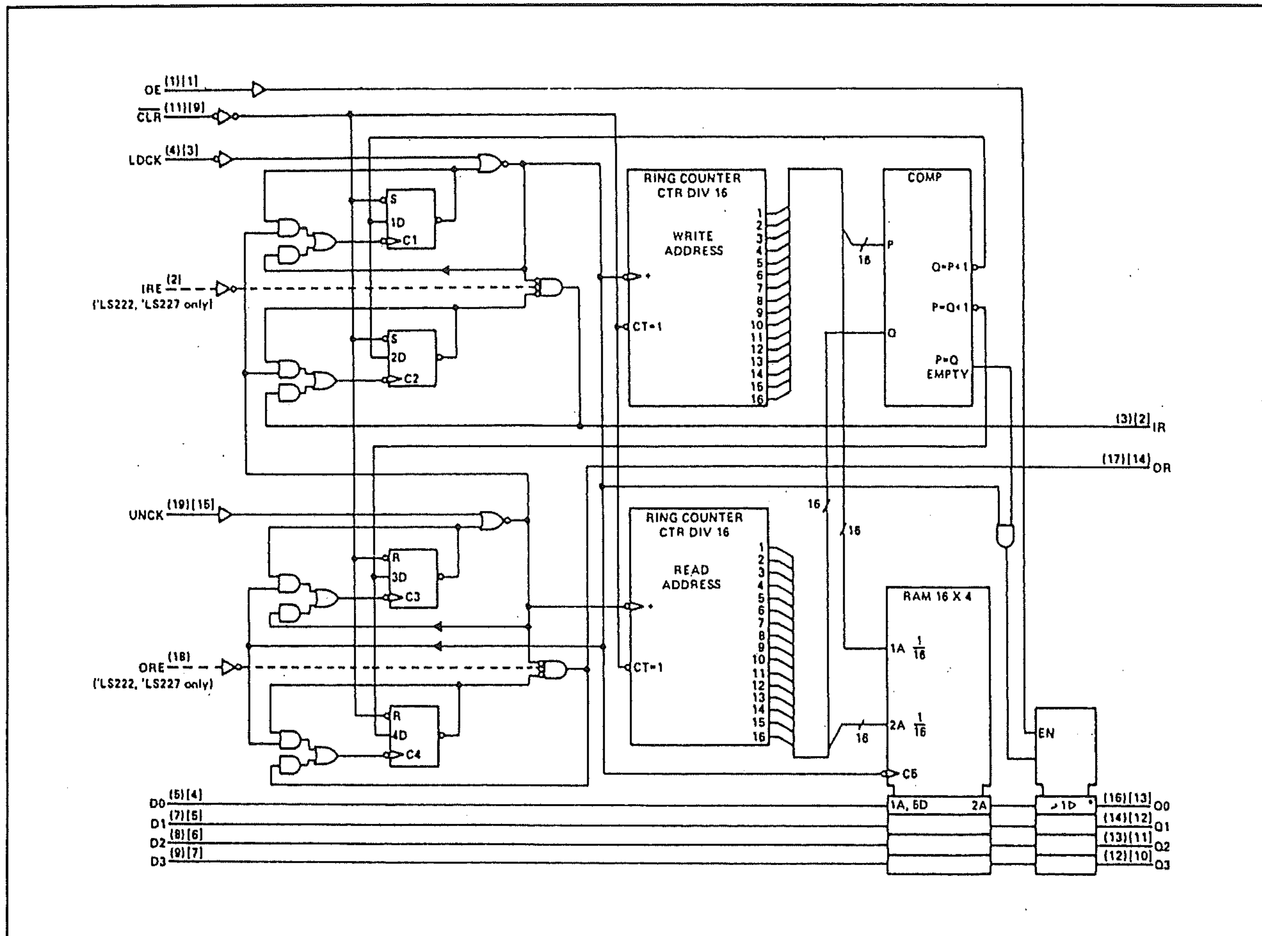


Figuur 3/6.27-5: Cirkelvormige geheugenstructuur van een FIFO met twee pointers.

Bovendien is na opstarten de informatie aan de uitgang pas geldig vanaf de N-de clockpuls, omdat de data daarvoor willekeurig en onbekend is. Deze FIFO heeft ook het nadeel dat niet met twee verschillende snelheden aan ingang en uitgang kan worden gewerkt. Een betere oplossing is naast de rij latches een reeks besturingseenheden op te nemen, zie figuur 3/6.27-3. De latches bevatten dan net als in figuur 3/6.27-2 de data. De besturingseenheden leveren nu "vlaggen" die aangeven of de bijbehorende latches geldige data bevatten of niet. Van bovenaf komende data kan dan telkens "zinken" tot in de latch die zich boven de laatst gevulde latch bevindt. Op deze wijze ontstaat dus een register met variabele lengte. In figuur 3/6.27-4 is te zien hoe een dergelijke FIFO er in de praktijk uitziet.

Een efficiëntere manier om een FIFO op te bouwen is gebruik te maken van een opslag-array dat zo breed is als de data en daarbij twee "pointers" toe te passen. De ene pointer wijst de locatie aan waar nieuwe data naar toe wordt geschreven, terwijl de tweede pointer bijhoudt waar de data moet worden uitgelezen.

6.27 Werking en principes van FIFO's

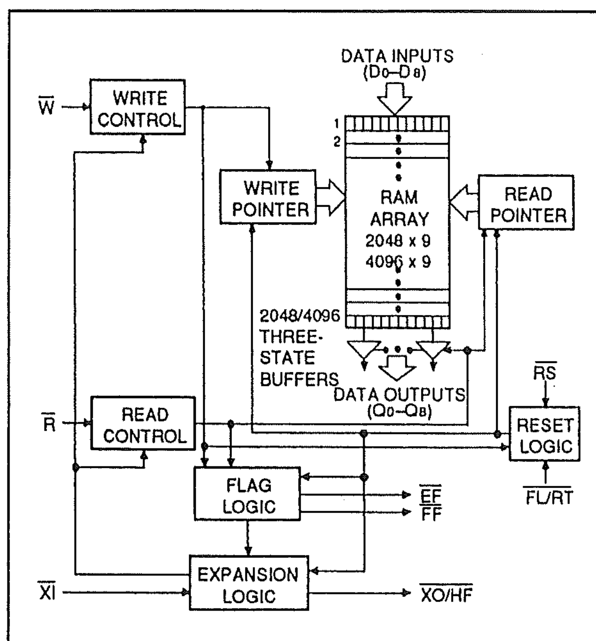


Figuur 3/6.27-6: Functioneel blokschema van een asynchrone FIFO (typen 74LS222/224/227/228) met ingebouwde RAM, ringtellers en comparator.

Wanneer één van beide wordt gebruikt om toegang tot een locatie te verkrijgen, wordt die automatisch met één verhoogd. Heeft een pointer de laatste positie in het array bereikt dan zal een volgende verhoging tot gevolg hebben dat de pointer op het begin van het array terecht komt. Deze structuur ziet er dus uit als een gesloten lus met twee pointers, figuur 3/6.27-5, en kan ook met software alleen worden uitgevoerd. Deze benadering resulteert in een veel kortere "doorvaltijd", terwijl de lengte toch variabel blijft. In figuur 3/6.27-6 is te zien hoe dit met een statische RAM wordt gedaan (alle onderdelen bevinden zich op één chip). De RAM is voorzien van een dubbele adressering: één voor de ingang en één voor de uitgang. Het is dus een "dual port" geheugen, echter met de beperking dat lezen en schrijven niet verwisselbaar zijn. De bovenste ringteller houdt het laadadres bij en de onderste het uitleesadres. De comparator vergelijkt beide adressen en levert daardoor het Input-Ready en het Output-Ready-signaal. Bij deze FIFO-typen wordt door een LAGE IR of OR ook de laad- of de ontlaadclock gesperd. De bovenste ringteller en de RAM worden tegelijk door het LDCK-signaal geklokt.

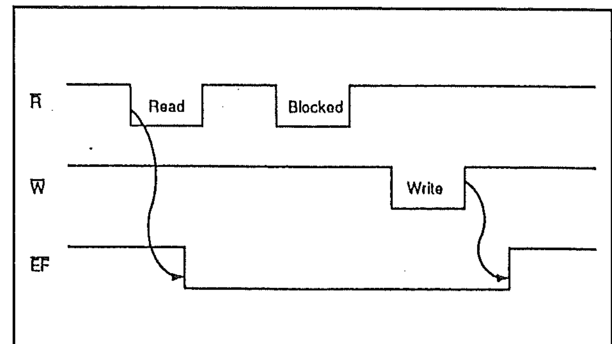
6.27 Werking en principes van FIFO's

Figuur 3/6.27-7 toont het vereenvoudigde blokschema van een moderne, grotere FIFO die volgens hetzelfde principe werkt. Er zijn nu drie statussignalen: FIFO leeg (empty: $\overline{EF}$) die met Output-Ready overeenkomt, half vol ($\overline{HF}$) en vol (full: $\overline{FF}$) die bij andere typen ook wel Input-Ready wordt genoemd. Met behulp van de $\overline{XI}$- en $\overline{XO}$-pennen kan de FIFO onbeperkt worden uitgebreid, terwijl de doorvaltijd 50 ns blijft.

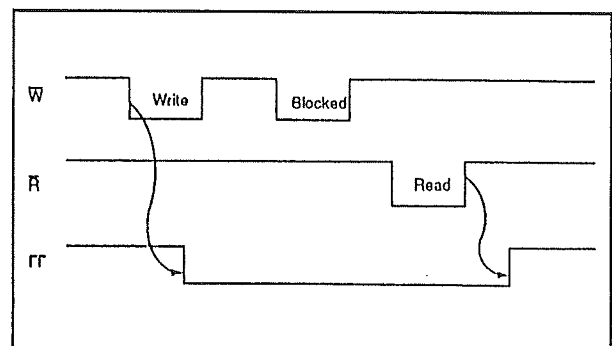


Figuur 3/6.27.7: Functioneel blokschema van een IDT7201/02 FIFO.

Na resetten wordt de leeg-vlag $\overline{EF}$ LAAG en zodra data wordt ingeschreven weer HOOG. $\overline{EF}$ gaat dan pas weer LAAG als alle informatie is uitgelezen, zie figuur 3/6.27-8. Als het aantal opgenomen data-elementen de helft van het totaal aantal plaatsen bereikt, gaat de halfvol-vlag $\overline{HF}$ LAAG. Zijn precies alle plaatsen in de FIFO van data voorzien, dan gaat de vol-vlag ($\overline{FF}$) LAAG om aan te geven dat er geen plaats meer is, zie figuur 3/6.27-9.



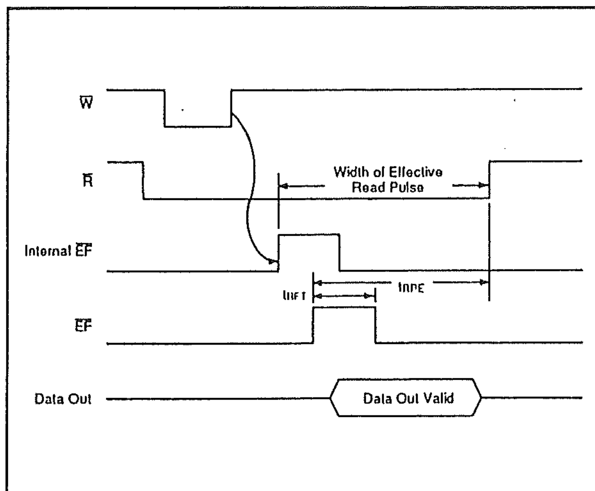
Figuur 3/6.27-8: Wanneer de FIFO slechts één woord bevat gaat de leeg-vlag ($\overline{EF}$) op de dalende flank van het leessignaal ($\overline{R}$) LAAG. Alle volgende leespulsen worden na het HOOG gaan van $\overline{R}$ geblokkeerd, terwijl $\overline{EF}$ LAAG blijft. Op de stijgende flank van de schrijfpuls $\overline{W}$ vervalt de leeg-conditie.



Figuur 3/6.27-9: Wanneer de FIFO op één woord na vol is, zal $\overline{FF}$ op de dalende flank van $\overline{W}$ LAAG gaan. Alle volgende schrijfpulsen worden na het HOOG gaan van $\overline{W}$ genegeerd, terwijl $\overline{FF}$ LAAG blijft. Op de stijgende flank van $\overline{R}$ wordt de vol-conditie opgeheven.

Wanneer de FIFO leeg is en $\overline{R}$ LAAG wordt gehouden voordat er schrijfpulsen LAAG gaan, vindt er automatische "Lees-doorstroming" (Read data flow-through)

plaats. De stijgende flank van $\overline{W}$ maakt $\overline{EF}$ HOOG, maar omdat $\overline{R}$ LAAG wordt gehouden treedt hierdoor een leescyclus op. Door dit lezen wordt $\overline{EF}$ weer LAAG (figuur 3/6.27-10). Op $\overline{EF}$ zullen zodoende HOOG gaande pulsen optreden met een breedte van minstens t_{RFT}. Op dezelfde wijze kan een "Schrijf-doorstroming" (Write data flow-through) optreden als de FIFO vol is en $\overline{W}$ LAAG wordt gehouden voordat een leespuls LAAG gaat.

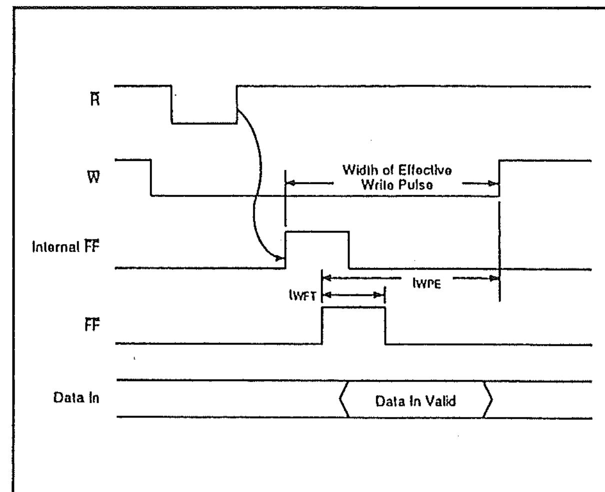


Figuur 3/6.27-10: Bij automatische lees-doorstroming wordt $\overline{R}$ LAAG gehouden.

De stijgende flank van $\overline{R}$ maakt $\overline{FF}$ HOOG, maar omdat $\overline{W}$ LAAG wordt gehouden treedt hierdoor een schrijfcyclus op. Dit schrijven maakt dat $\overline{FF}$ weer LAAG gaat (figuur 3/6.27-11). Op $\overline{FF}$ zullen dus pulsen HOOG gaan met een breedte van minstens t_{WFT}.

Bredere FIFO's

Wanneer in een toepassing FIFO's moeten worden gebruikt die breder zijn dan bestaande (of in voorraad gehouden) exemplaren, kunnen zij gemakkelijk worden verbreed door ze parallel te schakelen. In figuur 3/6.27-12 is een voorbeeld te zien van een FIFO voor 18 bit brede woorden. De status-vlaggen kunnen willekeurig van één van beide FIFO's worden gedetecteerd. Bij oudere typen kan externe logica nodig zijn om combinaties van Input-Ready en Output-Ready signalen te gebruiken (figuur 3/6.27-13).

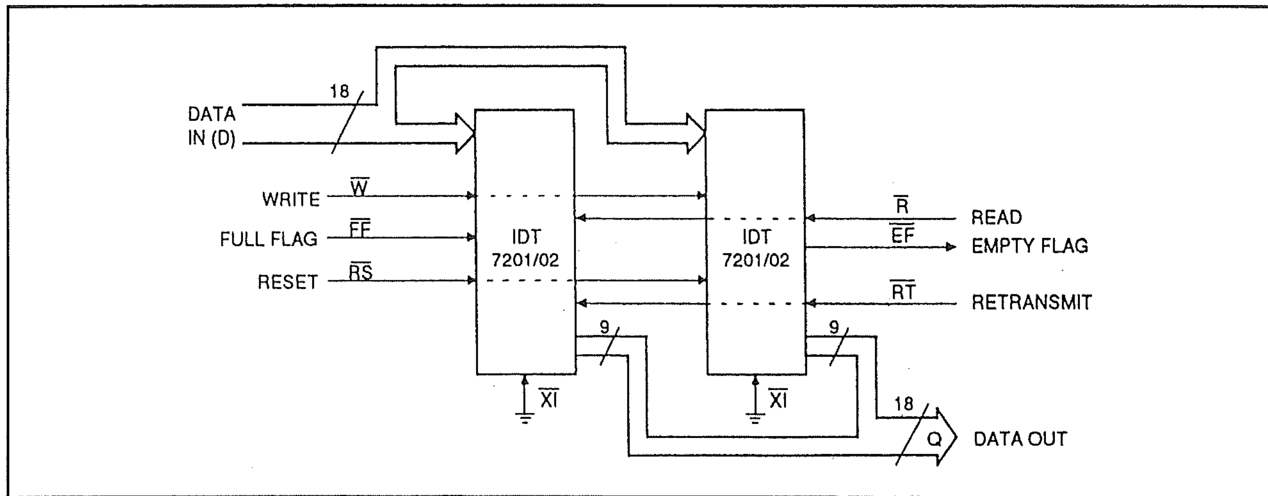


Figuur 3/6.27-11: Automatische schrijf-doorstroming treedt op als $\overline{W}$ LAAG wordt gehouden.

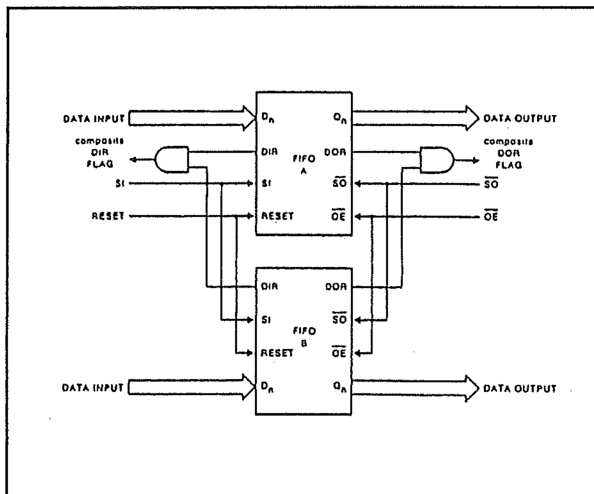
Langere FIFO's

Figuur 3/6.27-14 toont hoe eenvoudig de lengte (aantal woorden) bij sommige typen vergroot kan worden. De werking is als volgt. Data die met SIA door FIFO A wordt opgenomen komt na een vertraging aan de uitgang daarvan. Omdat $\overline{SOA}$ HOOG is wordt een DORA puls gegenereerd (=SIB). Na nog een vertraging komt de data op de uitgang van FIFO B. Dit gaat door totdat FIFO B vol is: DIRB (=$\overline{SOA}$) gaat LAAG, zodat geen SIB-pulsen meer worden opgewekt. De rest van de data komt verder alleen in FIFO A terecht. Bij uitlezen wordt SOB via DIRB aan FIFO A doorgegeven als SOA. Alle data schuift dus één plaats naar rechts en DIRA wordt HOOG.

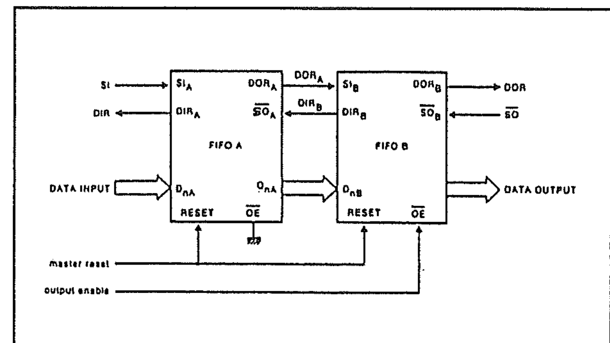
6.27 Werking en principes van FIFO's



Figuur 3/6.27-12: Uitbreiding van twee 7201/02-typen tot een FIFO met 18 bit woordbreedte.



Figuur 3/6.27-13: Bij sommige typen is voor grotere woordbreedten een combinatie van de DIR- en DOR- vlaggen nodig.

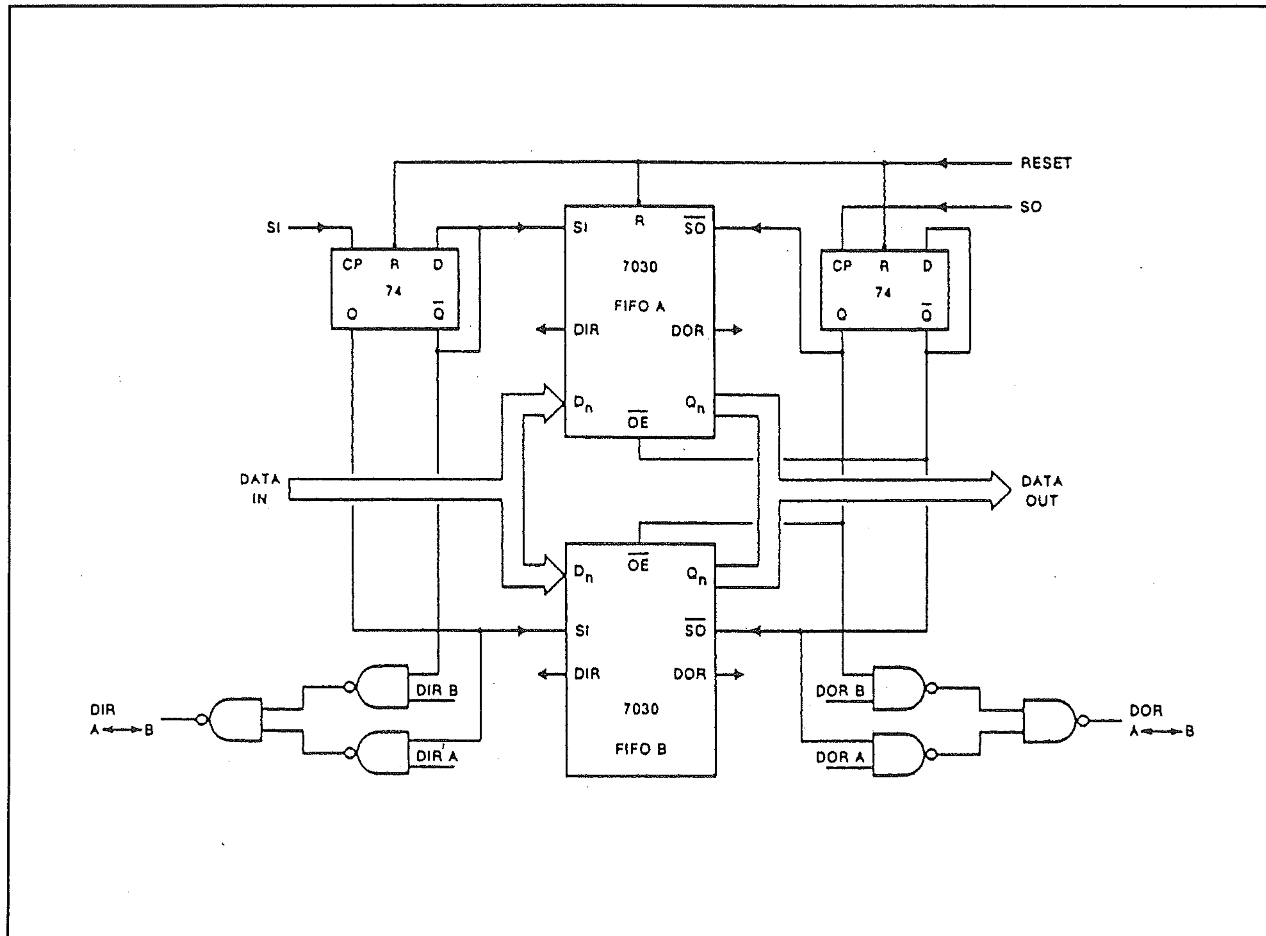


Figuur 3/6.27-14: Vergroting van het aantal woorden door in serie schakelen van twee (of meer) FIFO's.

Als FIFO A toevallig een langzaam type is en FIFO B een zeer snel type, kan het voorkomen dat de DIRB-puls te kort is om als $\overline{SO}$A-puls te dienen, zodat FIFO A niet goed werkt. Omgekeerd kan het ook gebeuren dat DORA te kort duurt om als SIB te werken (als FIFO B langzamer is dan FIFO A). In deze gevallen moeten de pulsen naar de andere FIFO worden verlengd.

Bij de schakeling zoals in figuur 3/6.27-14 neemt de doorvaltijd toe met het aantal FIFO's dat in serie wordt geschakeld. Het spreekt vanzelf dat bij zeer lange FIFO's de doorvaltijd dan ontoelaatbaar lang dreigt te worden. Een oplossing hiervoor is de FIFO's parallel te schakelen, zoals figuur 3/6.27-15 laat zien. De flip-flop's zorgen er dan voor dat beide FIFO's om de beurt worden gebruikt. Bij langere FIFO's moet natuurlijk een uitgebreidere decodering worden toegepast.

6.27 Werking en principes van FIFO's

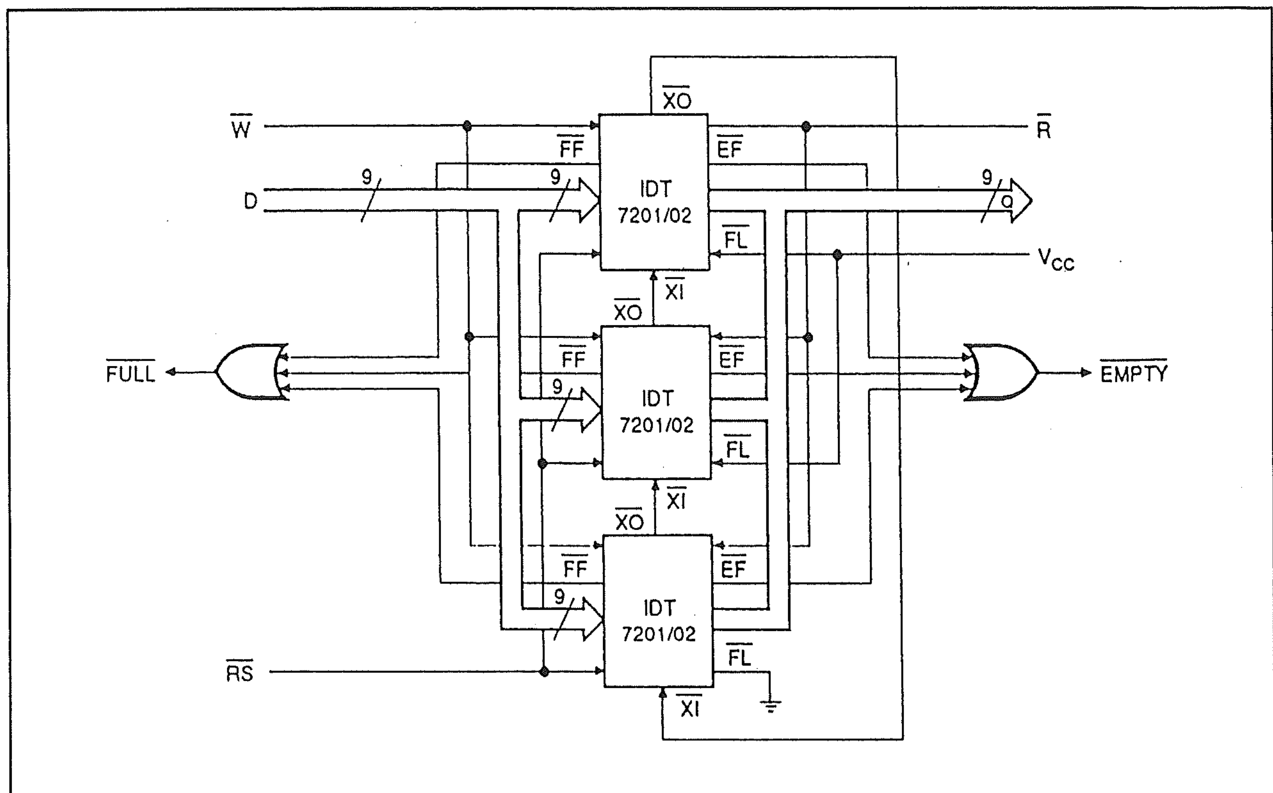


Figuur 3/6.27-15: Verdubbeling van het aantal woorden door middel van gemeenschappelijke data-in en data-uit bussen. Hierbij neemt de doorvaltijd niet toe.

Met twee-pointer FIFO's zoals de 7201 en 7202 kunnen alle data-ingangen met elkaar verbonden worden, evenals de data-uitgangen, zie figuur 3/6.27-16. Hierdoor ontstaat een parallelle architectuur die ook wordt gebruikt bij gewone RAM's om diepere geheugens te verkrijgen. Omdat FIFO's geen chip-selects en externe decoderingen kennen moet het kiezen van de juiste FIFO intern gebeuren. In de 7201/02 wordt dit bereikt door een unieke seriële structuur: de eerste (of master) FIFO wordt geïdentificeerd door de $\overline{FL}$-ingang te aarden. Van alle overige FIFO's in de schakeling moet de $\overline{FL}$- ingang worden opgetrokken naar V_{CC}. De $\overline{XO}$-uitgang van de eerste FIFO wordt verbonden met de $\overline{XI}$-ingang van de volgende FIFO in de rij, enzovoort. De $\overline{XO}$-uitgang van de laatste FIFO wordt tenslotte aangesloten op de $\overline{XI}$-ingang van de eerste.

Na het resetten staan de actieve lees- en schrijf-pointers R en W op de eerste FIFO. Als de schrijf-pointer (W) aan het einde van de eerste FIFO is gekomen ontstaat een puls op de $\overline{XO}$-uitgang, waardoor de schrijf-pointer aan het begin van de tweede FIFO wordt geactiveerd, terwijl tegelijkertijd de schrijf-pointer van de eerste wordt uitgeschakeld.

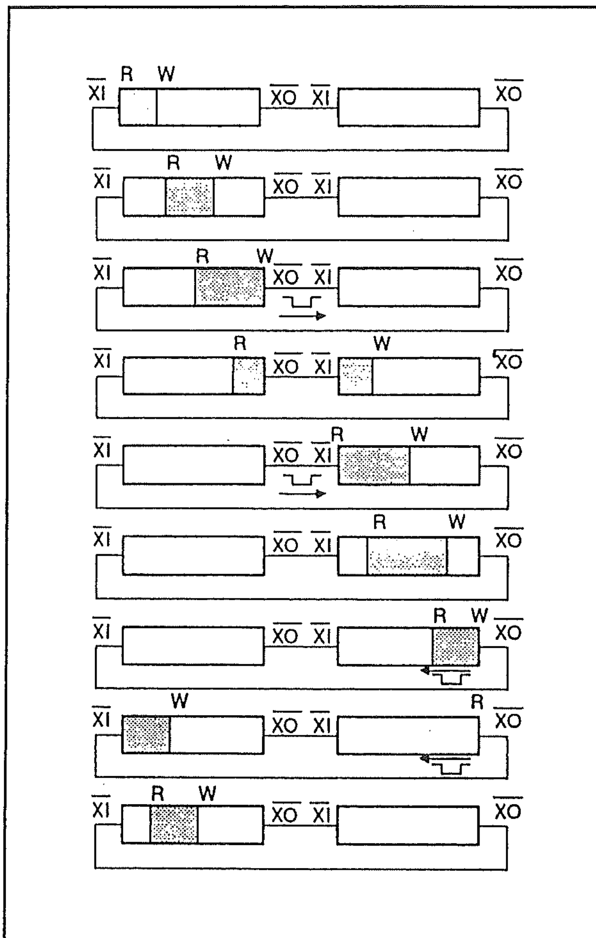
6.27 Werking en principes van FIFO's



Figuur 3/6.27-16: Opbouw van een snelle, langere FIFO waarbij de $\overline{XI}$- en $\overline{XO}$-lijnen worden gebruikt. De data-ingangen zijn met elkaar verbonden, evenals de data-uitgangen.

De besturing van Write-Enable wordt zodoende doorgegeven naar de tweede FIFO. Wanneer de actieve lees-pointer (R) het einde van de eerste FIFO bereikt activeert hij via een tweede puls op de $\overline{XO}$-uitgang de lees-pointer in de tweede FIFO en schakelt hij zichzelf uit. In figuur 3/6.27-17 is de verplaatsing van de pointers met behulp van $\overline{XO}$ en $\overline{XI}$ over twee FIFO's te zien. In deze ringstructuur loopt de lees-pointer altijd na op de schrijfpointer (hij kan deze niet passeren).
In figuur 3/6.27-18 wordt tenslotte getoond hoe de positie van de pointers is wanneer een grotere hoeveelheid data wordt opgenomen. Alleen na een reset bevinden beide pointers zich voorin de eerste FIFO.

Nadat alle data is uitgelezen staan W en R weer tegenover elkaar, maar dan op een willekeurige plaats.
De werking is als volgt:
- A:
  na een master-reset staan W en R aan het begin van FIFO 1;
- B:
  FIFO 1 is gevuld en resterende data komt in FIFO 2 terecht;
- C:
  FIFO 1 is leeggelezen, het schrijven gaat door;
- D:
  de leeggelezen FIFO 1 wordt ook beschreven (wrapped around).



Figuur 3/6.27-17: De verplaatsing van de lees- en schrijfpointers over twee FIFO's. Telkens wanneer W of R de grens van een FIFO overschrijdt wordt een $\overline{XO}$-puls gegenereerd.

Software of hardware?

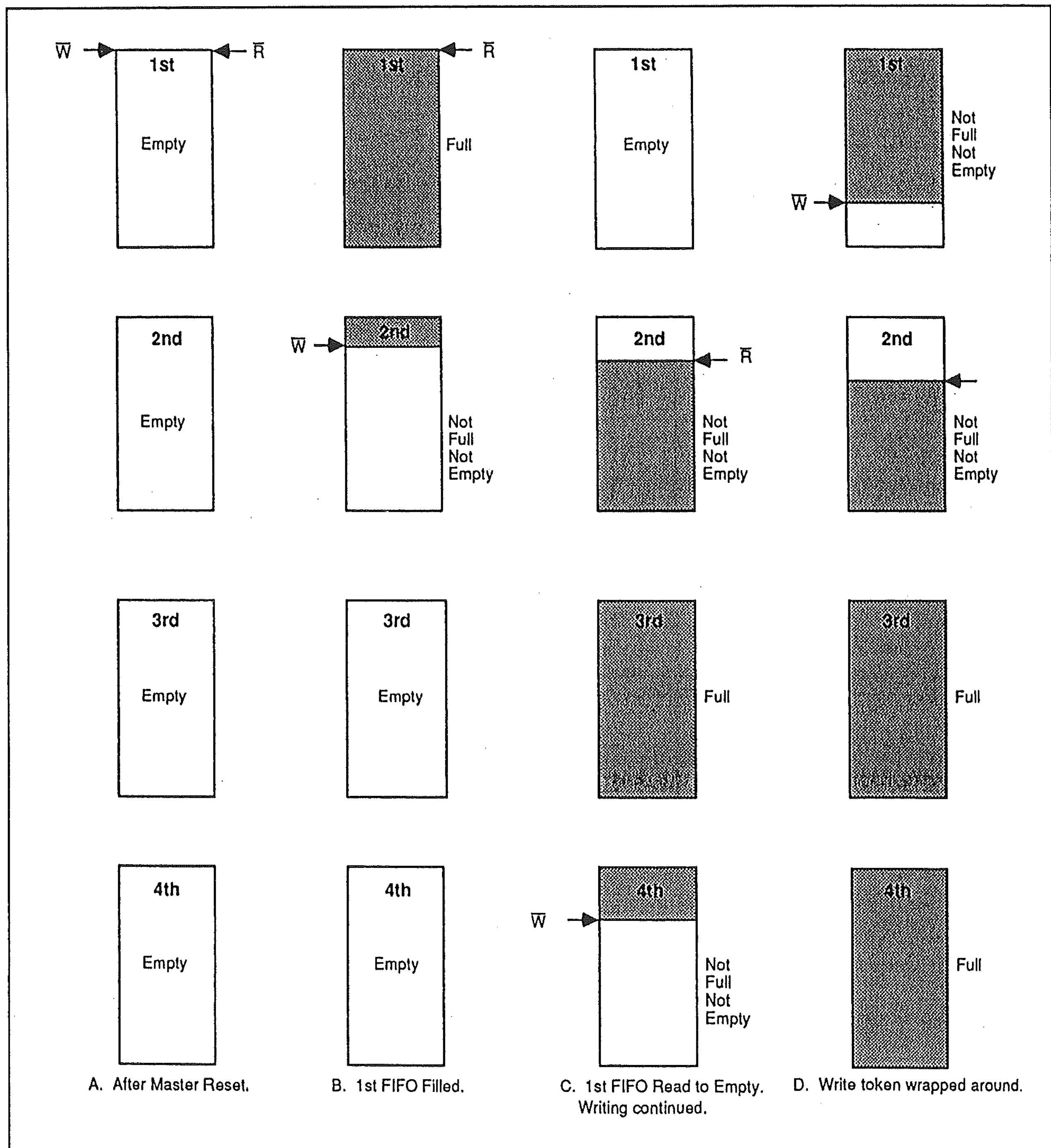
Inleiding

Bij iedere toepassing van een computer of programmeerbare besturing moet de ontwerper beslissen of bepaalde functies in software of in hardware worden uitgevoerd. Over het algemeen is de software oplossing flexibeler (kan gemakkelijker worden gewijzigd). De taak wordt echter langzamer uitgevoerd. De hardware oplossing is minder flexibel, maar doet de taak wel snel.

File-server

Als eerste voorbeeld wordt een file-server besproken. De server kan aan de ene zijde op een Local Area Network (LAN) zijn aangesloten en aan de andere zijde op een Winchester disk drive. Beide I/O-verbindingen hebben op onvoorspelbare momenten aandacht nodig en moeten dan direct worden bediend omdat anders data verloren gaat. Als de datasnelheid van beide interfaces laag genoeg is, kan een totale software oplossing worden overwogen. De status van elke I/O-poort moet dan op tijd worden afgevraagd en data moet in volgorde van binnenkomst in een software FIFO-array worden opgeslagen (figuur 3/6.27-19).
Terwijl de data wordt verwerkt moeten beide interfaces evengoed in de gaten gehouden worden. Het is twijfelachtig dat een dergelijke oplossing betrouwbaar werkt. Een betere manier is om interrupts door beide interfaces in te voeren. Hierbij kan dan een taak op volle snelheid worden uitgevoerd, terwijl bijna direct op een I/O-routine kan worden overgeschakeld (figuur 3/6.27-20). Omdat de interrupts op elk willekeurig moment kunnen optreden, moeten de interrupt-service routines zo worden ontworpen dat zij geen data vernietigen die toevallig door de geïnterrumpeerde taak wordt gebruikt. De routines moeten de toestand van de machine zorgvuldig opbergen, hun taak uitvoeren en de toestand van de machine opnieuw invoeren. De extra code voor de machinetoestand is teveel informatie die in het ergste geval niet op tijd verwerkt kan worden. Op kritieke punten moet de ontwerper dan code tussenvoegen om interrupts te kunnen tegenhouden.

6.27 Werking en principes van FIFO's

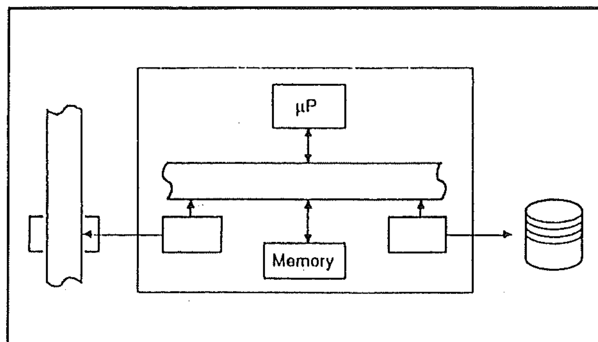


Figuur 3/6.27-18: Positie van data en pointers in vier FIFO's die op de manier van figuur 3/6.27-16 zijn geschakeld.

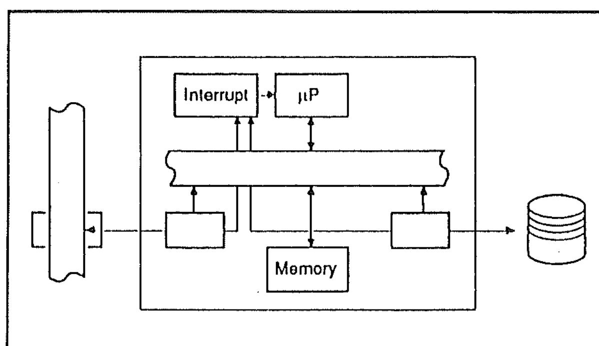
De interrupt-oplossing kan wat meer naar de hardware-kant worden verplaatst door de DMA-techniek (Direct Memory Access) toe te passen (figuur 3/6.27-21). Hierbij worden de I/O-poorten door een aparte schakeling in de gaten gehouden. Als een poort dan attentie vraagt, interrumpeert de DMA-logica de lopende taak

6.27 Werking en principes van FIFO's

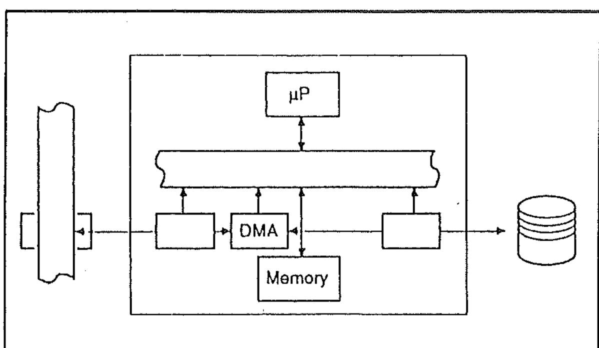
op bustransfer niveau en wordt een geheugencyclus gestolen om de data van de poort naar de FIFO-array in het geheugen te brengen (of andersom).



Figuur 3/6.27-19: Software implementatie van een file-server.

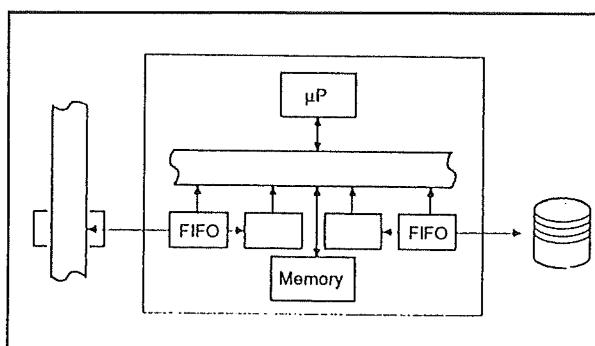


Figuur 3/6.27-20: Interrupt implementatie van een file-server.



Figuur 3/6.27-21: De oplossing met DMA naar de processor.

De lopende taak mist hierdoor slechts af en toe enkele geheugencycli. De DMA-oplossing is echter niet gratis. DMA-controllers zijn complexe schakelingen die voor een bepaalde busstructuur zowel geprogrammeerd als ontworpen moeten zijn.
Bovendien kan het DMA-mechanisme slechts één bron tegelijk bedienen en vormt daardoor toch nog een belemmering. De beste oplossing blijkt dus een hardware FIFO te zijn (figuur 3/6.27-22). De periferie kan hierbij de data laden of inlezen zonder de besturing te interrumperen. Aangezien de besturing niet betrokken is bij het in stand houden van de datastromen, kan ook geen data verloren gaan.

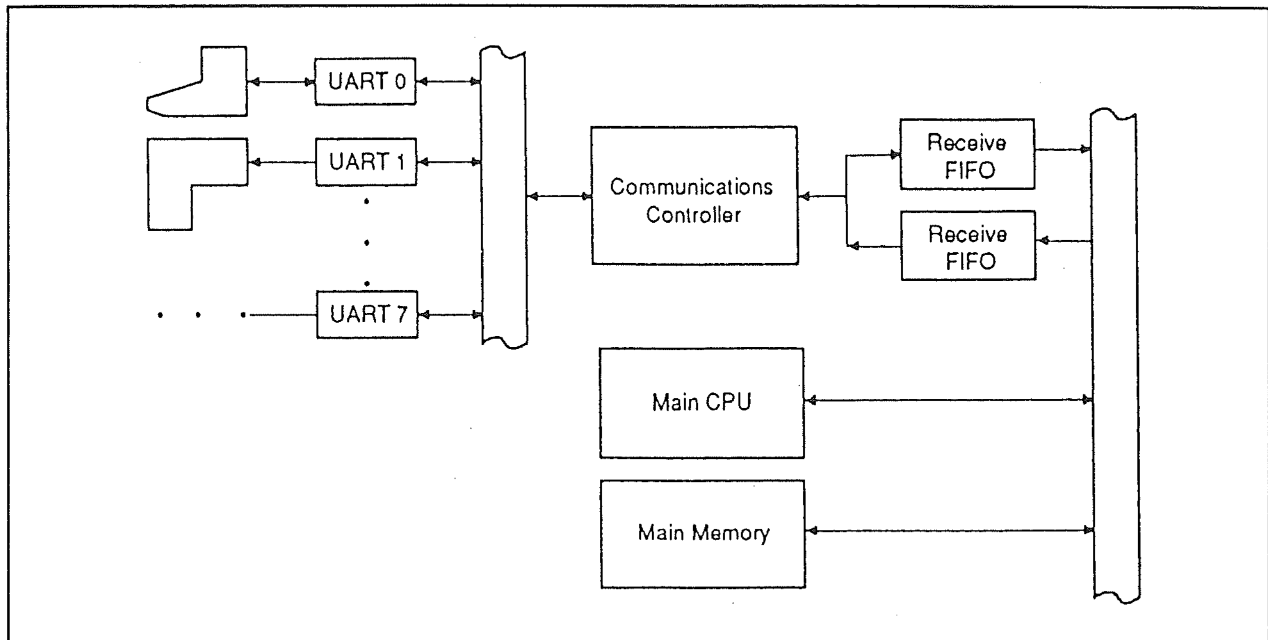


Figuur 3/6.27-22: Implementeren van een file-server met FIFO's.

Communicatie-controller

Een ander voorbeeld is een communicatie-controller. Deze dient om de CPU te verlossen van het in de gaten houden van meerdere UART's van verschillende terminals en printers.
Zoals in figuur 3/6.27-23 te zien is, kan deze controller dienen als communicatie-multiplexer en data-concentrator. Hierbij wordt zowel aan zend- als aan ontvangstzijde een FIFO gebruikt.

6.27 Werking en principes van FIFO's



Figuur 3/6.27-23: Een communicatie-controller met FIFO's om datastromen in beide richtingen op te vangen.

Leverbare FIFO's

Inleiding

In de 74xx- en (1)4xx-series worden diverse FIFO's aangeboden, die nu even worden samengevat.

74(LS)222	16 x 4 bit, 3-state, OR-enable
74(LS)224	16 x 4 bit, 3-state
74(S)225	16 x 5 bit, 3-state
74(LS)227	16 x 4 bit, open-collector OR-enable
74(LS)228	16 x 4 bit, open-collector
74(HCT)7030	64 x 9 bit, 3-state
74(HCT)7403	64 x 4 bit, 3-state
74(HCT)7404	64 x 5 bit, 3-state
74(HCT)40105	16 x 4 bit, 3-state
74(ALS)229A	16 x 5 bit FIFO
74(ALS)232A	16 x 4 bit FIFO
74(ALS)233A	16 x 5 bit FIFO
74(ALS)234	64 x 4 bit FIFO
74(ALS)235	64 x 5 bit FIFO
74(ALS)236	64 x 4 bit FIFO
74(ACT)7201A	512 x 9 bit FIFO
74(ACT)7202	1024 x 9 bit FIFO
74(ALS)2232	64 x 8 bit FIFO
74(ALS)2233	64 x 9 bit FIFO
(1)40105	16 x 4 bit, 3-state

6.27 Werking en principes van FIFO's